

Tracking of Multiple Merging and Splitting Targets: A Statistical Perspective ¹

Curtis B. Storlie², Thomas C. M. Lee³, Jan Hannig⁴, and Douglas Nychka⁵

Date: June 6, 2006; first revision: April 6, 2007; second revision: November 2, 2007

Abstract

This article considers the important problem of tracking multiple moving targets captured in image sequences. It has two primary objectives. The first one is to serve as an introduction of the target tracking problem to the statistical community. It achieves this by providing a common definition of the tracking problem, a survey of important existing work, and a discussion of the relative advantages and shortcomings of such work. The second objective is to propose a statistical method for solving a wide class of tracking problems, namely, when the system of interest contains birth, death, merging and splitting of targets. The stochastic model behind this method is continuous time in nature and is equipped with a realistic mechanism for handling merging and splitting. Its finite sample properties are assessed via numerical experiments. Finally, the method is applied to two real scientific problems for which it was originally designed for: the tracking of (i) storms captured in radar reflectivity image data and (ii) vortexes from a high-resolution simulated vorticity field.

Keywords: Multiple Target Tracking, Merging, Splitting, Multiple Hypothesis Tracking, Track Estimation, Convective Systems, Turbulence.

1 Introduction

Multiple target tracking is an important problem arising in many scientific and engineering investigations. It has importance in radar and signal processing, air traffic control, robot vision, GPS-based navigation, biomedical engineering, and video surveillance to name a few. In this article we provide an introduction to and propose a new statistical method for multiple target tracking. Our work is motivated by the scientific need of storm tracking from radar reflectivity data and vortex tracking in turbulence fields. In these applications, the splitting and merging of targets are

¹Running Title: Multiple Target Tracking

²Corresponding Author. Department of Mathematics & Statistics, University of New Mexico. MSC03 2150, 1 University of New Mexico, Albuquerque, New Mexico 87131-0001 Email: storlie@stat.unm.edu

³Department of Statistics, The Chinese University of Hong Kong, and Department of Statistics, Colorado State University. Email: tlee@sta.cuhk.edu.hk

⁴Department of Statistics, Colorado State University. Email: hannig@stat.colostate.edu

⁵Geophysical Statistics Project, National Center for Atmospheric Research. Email: nychka@ucar.edu

quite common, and such events are effectively and realistically accommodated by the proposed method. It is also our hope that this article will stimulate more statistical work to be conducted in the interesting and exciting area of target tracking.

1.1 Problem Definition

Typically a complete tracking application is composed of two parts. The first part is to extract the locations and/or other attributes of the targets from each image frame. There is no unified solution for this, as different targets need different methods for extraction. For example, human faces and missiles require very different target recognition methods to detect their appearances in an image. Once the target coordinates are located, the second part of the tracking application is to link these coordinates together so that coordinates of the same target detected at different image frames are connected to form a reconstruction of the path that this target traveled. In the tracking literature this second part of coordinate linking is commonly referred as the *data association problem*.

For the rest of this article we assume that the target coordinates and/or other useful attributes have already been extracted from the image sequence and focus on the second step, the data association. However, we do **not** assume that these targets are perfectly extracted. That is, we allow some of the real targets to remain undetected in some image frames (i.e., “*missing*” targets), we allow the presence of false alarms, and we also allow the target coordinates to be recorded with measurement errors. Furthermore, we also assume the occurrences of the following four events. First, we allow targets to appear for the first time or disappear permanently at any times during the image sequence. These two events are called *birth* and *death* respectively. Second, we permit situations under which two targets combine together to form a larger target; this is called *merging*. Lastly, targets are also allowed to break into smaller pieces; this is called *splitting*.

In summary, the tracking problem that this article considers is to, given the target information (coordinates and/or other attributes) acquired by the first extraction phase, recover the path, also known as *track*, of each target traveled. In doing so we allow birth, death, merging and splitting of targets. We also allow missing targets and false alarms. Our approach to this tracking problem is to first fit a stochastic model that incorporates all variables of interest, including times of birth,

death, merging and splitting events, as well as target locations. We then estimate their conditional distribution given the data. Finally, the solution that corresponds to the mode of this estimated distribution is taken as the tracking estimate.

1.2 Review of the Tracking Problem

We first briefly review some popular methods for tackling the above tracking problem. The reader is also referred to the books by Bar-Shalom, Li & Kirubarajan (2001), Blackman & Popoli (1999), and Stone, Barlow & Corwin (1999) for a comprehensive description of modern tracking techniques.

The target tracking problem has been studied extensively in the engineering literature over the past thirty years. The approaches to this problem can be loosely classified into two groups: non-statistical and statistical. The non-statistical group mainly use either image differencing techniques to detect target movements for consecutive images as in Pece (2002), or heuristically minimize various objective functions that penalize the smoothness of the track estimates (e.g., Sethi & Jain 1987 and Salari & Sethi 1990). Typically these non-statistical methods are fast and simple, and have been used with some success in the area of storm tracking (e.g., Johnson, Mackeen, Witt, Mitchell, Stumpf, Eilts & Thomas 1998, Wolfson, Forman, Hallowell & Moore 1999, Dixon 1994, Tuttle & Gall 1999, Lakshmanan, Rabin & DeBrunner 2003, Hodges 1994, and Hodges 1999). However, they also possess a serious drawback: their inability to adequately handle birth, death, missing targets and false alarms. Moreover, these methods are only able to provide track estimates of the target movements and do not offer any mechanisms for assessing the associated uncertainties.

Hence, for the rest of this article, we shall focus our attention on the statistical approaches. The main idea is to employ a statistical model to describe the movements of the targets. Once a target model is proposed, the traditional approach to the data association problem is to then find the collection of tracks that maximizes the likelihood of the model. In the tracking literature such a collection of tracks is often called a *hypothesis*. To achieve such a likelihood maximization, two major issues are involved: (i) fast calculation of the likelihood value for any given hypothesis and (ii) an effective search algorithm for locating the hypothesis that maximizes the likelihood function.

In the literature typically a linear Gaussian state space model is applied to address the first

issue, as it allows for efficient likelihood calculation of a given data association hypothesis via the Kalman filter. However, methods have also been developed for efficient likelihood calculation for more general cases. For examples, the extended Kalman filter (Anderson & Moore 1979) is based on local linear approximations to a nonlinear system, and the unscented Kalman filter (Julier & Uhlmann 2004) uses higher order approximation to calculate likelihood for nonlinear systems.

Most recently, research in this area has focused on a new class of filtering methods based on particle filtering or sequential Monte Carlo (Gordon, Salmond & Smith 1993, Kitagawa 1996, Liu & Chen 1998, Doucet, Godsill & Andrieu 2000 and Doucet, de Freitas & Gordon 2001). This approach represents the target distribution with a set of samples, called particles, each has its own importance weight. These particles are then propagated through time to provide an approximation to the distribution at subsequent time steps. Liu & Chen (1998) develop a sequential importance sampling framework under which most of these procedures can be unified. Additionally Chen & Liu (2000) consider a class of models that are conditionally linear and Gaussian. Still other methods such as the probability hypothesis density method (Mahler 2003 and Vo, Singh & Doucet 2005) approximate the likelihood by propagating only the posterior expectation instead of the entire distribution through to subsequent times.

Assuming a method for fast likelihood calculation is available, the next step is to develop a search algorithm to find the data association hypothesis that maximizes the likelihood of the model. The most widely used heuristic search algorithm for this purpose is the Multiple Hypothesis Tracking (MHT) algorithm of Reid (1979). A brief description is given in Section 4.3. We have implemented a variant of this MHT algorithm for our likelihood maximization; see also Section 4.3.

There is also the Bayesian approach to the data association problem where a prior distribution is given to the possible data associations. Notice that in this setup the prior is imposed on the data associations but not on any model parameters. Usually this prior is an uninformative point uniform distribution over all of the possible associations. This is akin to assuming that the observations of targets/false alarms at each time are recorded in a random order. The solution is then to calculate the posterior distribution of the data association hypotheses given the observations. Since this is a very computationally intensive task, standard MCMC methods are generally too time consuming for

many applications. To overcome this issue, several particle filter approaches have been suggested to take advantage of the sequential nature of this problem (Särkkä, Vehtari & Lampinen 2007, Kreucher, Kastella & Hero 2005, and Vermaak, Godsill & Perez 2005). In Section 4 we have adopted an MHT approach to approximating this posterior distribution. This is similar in spirit to the ideas presented in Obermeyer & Poore (2004).

As mentioned earlier, this article is motivated by the scientific need of tracking merging and splitting targets, such as storms or vortexes. Merging and splitting of targets can be common in radar applications as well, though in a slightly different context. That is, when two targets are close together, resolution limits may prevent them from being simultaneously detected. The detection method will then return only one (or even no) observation for these two targets. This certainly poses additional difficulties and challenges. Although this is perceived as a very important issue by Daum (1994) and Blackman (2004), we are unaware of any satisfactory solution to this. Most existing methods for tracking merging targets are not well defined in terms of an overall probabilistic model (Trunk & Wilson 1981, Chang & Bar-Shalom 1984, Koch & van Keuk 1997 and Genovesio & Olivo-Marin 2004).

Up to now the most complete model for merging and splitting targets seems to be the one presented in Khan, Balch & Dellaert (2005) and Khan, Balch & Dellaert (2006). This model is appropriate for the situation of unresolved radar measurements discussed above. However, for the following reasons, this model is inadequate for representing the physical processes that we are studying in Section 2. First, it does not allow for the birth and death of targets. Second, no dependence structure is specified to describe the relationship between the merging events and the corresponding parent target locations. In other words, the model would allow for merging of targets that are distant apart. Lastly, merging and splitting events are very temporary, in the sense that newly merged targets are equally likely as any other targets to not be merged in the next image frame. This is unrealistic for many real problems as when targets merge, they tend to stay merged for an extended period of time. Therefore, there is a clear need for a tracking method that is designed to handle all the above issues satisfactorily.

1.3 The Proposed Method

A major goal of this article is to develop a statistical tracking method that addresses all of the concerns listed in the preceding paragraph. This method is based on a new continuous time stochastic model that incorporates birth, death, merging and splitting of targets into the likelihood, as well as missing targets and false alarms. It also allows us to accommodate other target attributes (such as size or intensity) to improve the estimation results. Utilizing attribute information has been shown to increase tracking performance in other applications (e.g., Bal & Alam 2005, Roh, Kang & Lee 2000, Salmond & Parr 2003, and Angelova & Mihaylova 2006). To the best of our knowledge, this is the first time all the above events are explicitly built into a stochastic model for tracking. We note that, although this model is fairly complex and contains different components, it was designed in such a way that the resulting likelihood function can still be expressed in closed form and hence can be computed efficiently.

The proposed model is continuous in time which provides two additional advantages. First, it can be easily applied to irregularly sampled image sequences, and secondly, it allows the asymptotic properties of the tracking estimates to be studied when the sampling time converges to zero. Although it is beyond the scope of this article to examine these large sample properties, a theoretical justification for our tracking method is provided in Storlie, Hannig & Lee (2007).

When comparing to existing methods, another advancement of our work is the way that we handle the static model parameters (e.g. rates of birth and death, noise variances, etc.). There are many approaches to particle filtering that allow for estimation of static model parameters along with the state variables (e.g., Andrieu & Doucet 2003, Doucet, de Freitas & Gordon 2003 and Doucet et al. 2001). However, this is rarely applied to multiple target tracking problems, for which most existing methods assume that many of the key model parameters are known. We provide consistent estimates of these model parameters and we calculate the distribution of the data associations given these estimates and the observations. Our approach might therefore be considered empirical Bayes where, just like the Bayesian approach to tracking, an uniform prior is imposed on the data associations but not on any of the model parameters.

The rest of this article is organized as follows. In Section 2 we describe two scientific prob-

lems that motivate our work. The proposed stochastic model is presented in Section 3. Section 4 demonstrates how this model can be applied to provide track estimates of the targets. Finite sample performance of the methodology is then illustrated on simulated data in Section 5. Sections 6 and 7 report the tracking results obtained by applying the method to the two original scientific problems. Lastly concluding remarks and possible future work are given in Section 8. We remark that there is also an online supplementary document to this article, available at <http://www.stat.colostate.edu/~tlee/tracking/>. This document provides additional information that we will refer to. These references are always preceded by the letter S. For example, equation (S.1) refers to the first equation in the supplementary document.

2 Two Motivating Problems

This section describes two scientific problems for which the tracking of multiple targets is an important step to their solutions. It should be noted that these examples are nonstandard tracking problems which are meant to serve as a clear illustration of our methodology. However, this results in a difficulty level that is not necessarily representative of current state-of-the-art tracking challenges.

2.1 Convective Systems

Figure 1 shows radar reflectivity images evolving over time on July 14, 1996 from 1:00am to 3:30am. Radar reflectivity is correlated to rainfall intensity so that we can roughly attribute the variation of color in these images to different rainfall activities. In these images blue indicates 0 inches/hour of rainfall increasing on a log scale to bright yellow indicating ≥ 2 inches/hour. The images are separated by 30 minutes.

The targets that we wish to track are the larger convective systems. For our purposes, a convective system is defined to be a rainfall system that is larger than 100 km in length (approximately 1° of latitude or longitude). This problem has been studied previously by Davis, Manning, Carbone, Trier & Tuttle (2003). The very short term behavior (less than 1 hour) of such systems are reasonably well known, but the moderately short term (1 to 6 hours) and long term (1 to 2 days)

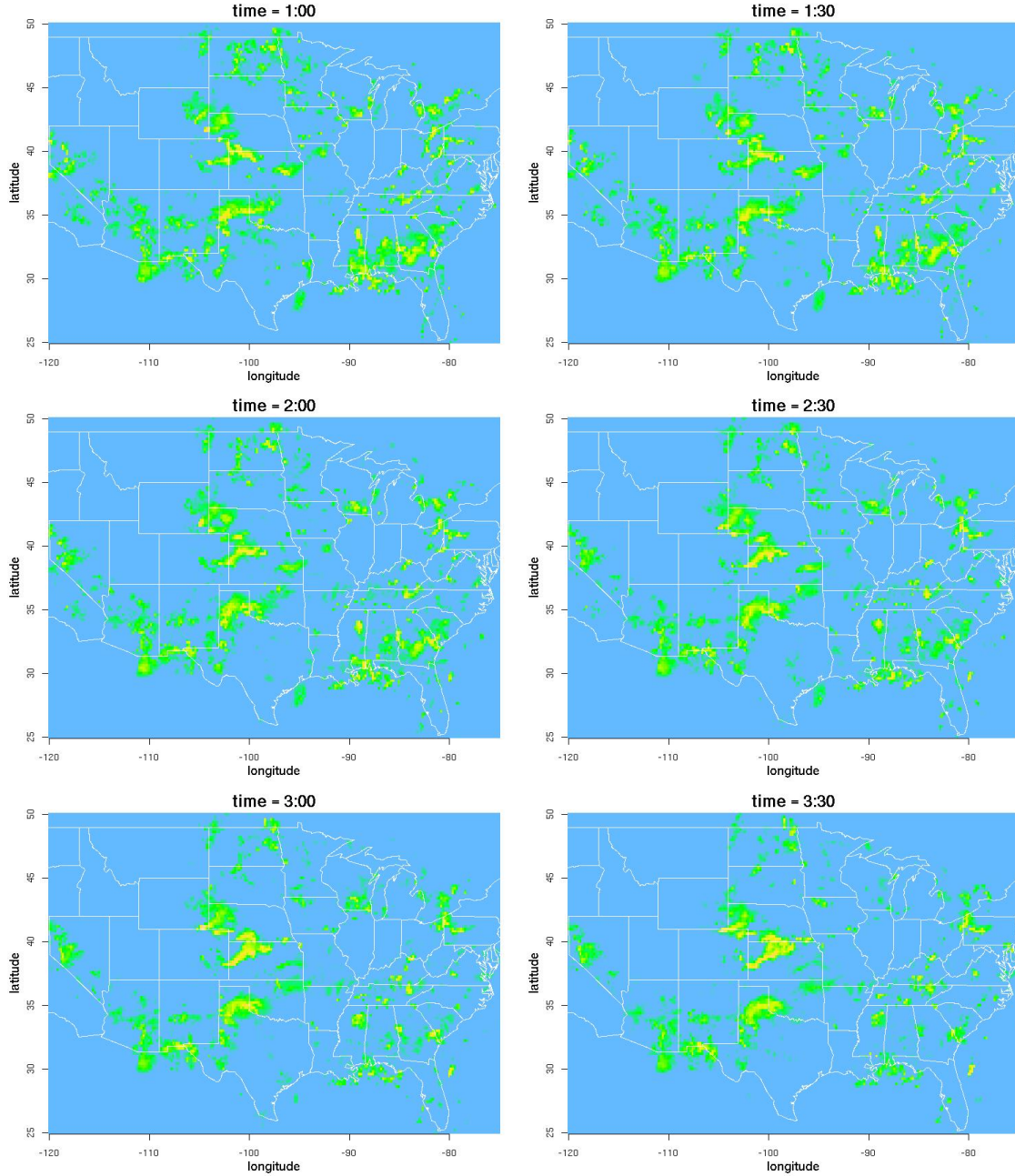


Figure 1: Radar Reflectivity Images Evolving During July 14, 1996

behaviors are still largely unknown. It is certainly desirable for the purposes of climate modeling to gain a better understanding of the longer term behavior.

A real scientific contribution that can be made then is the validation and improvement of the storm activity in Regional Climate Models. These are complex computer models for the state of the atmosphere. These models are used for the prediction of weather and also to generate data

to answer scientific hypotheses. It is thus necessary for these models to maintain a high degree of reality in terms of the storms that they produce. Comparing the distribution of storm tracks from real data to that of the Regional Climate Model is one way to verify this. Hence, a much needed tool is a procedure that can recover the movements and the interactions of all of the convective systems.

The merger of two systems that are located at the corner of South Dakota, Wyoming, and Nebraska can be seen in the first few images. These are clearly separate systems until the third image, where they are now one system. There is also a splitting event in the panhandle of Florida as a fairly large system breaks apart into two smaller systems.

2.2 2D Turbulence

The second motivating problem focuses on a 2D turbulence simulation of freely decaying vortexes; see Figure 2. These images are of a 2D vorticity field with random initial conditions as it develops over time with no energy loss to the overall system. The white objects are centers of vorticity rotating in a clockwise direction, whereas the black vortexes have the opposite rotation.

Vortexes of the same spin will coalesce as they move close to each other. There is a good example of a merger between times 8 and 9 where two white vortexes that are left of center and below center in the images. Also, vortexes of opposite spin have a tendency to parallel each other for a while before moving off in different directions. Two vortexes exhibiting this behavior are called dipoles. An example of dipoling are the two vortexes that are left of center and above center in the images. They both travel upwards and slightly left during the image sequence.

Recently such image sequences are a subject of much research (Bracco, McWilliams, Murante, Provenzale & Weiss 2000, Pasquero, Provenazale & Weiss 2002, and Weiss & McWilliams 1993), as it is a paradigm for anisotropic geophysical and astrophysical turbulence and at the same time it is also the most computationally accessible example of fluid turbulence. Turbulence remains a largely open area of research. Automatic tracking of turbulence structures in this simple example is an important first step to achieve a better understanding of turbulence dynamics in more complex systems.

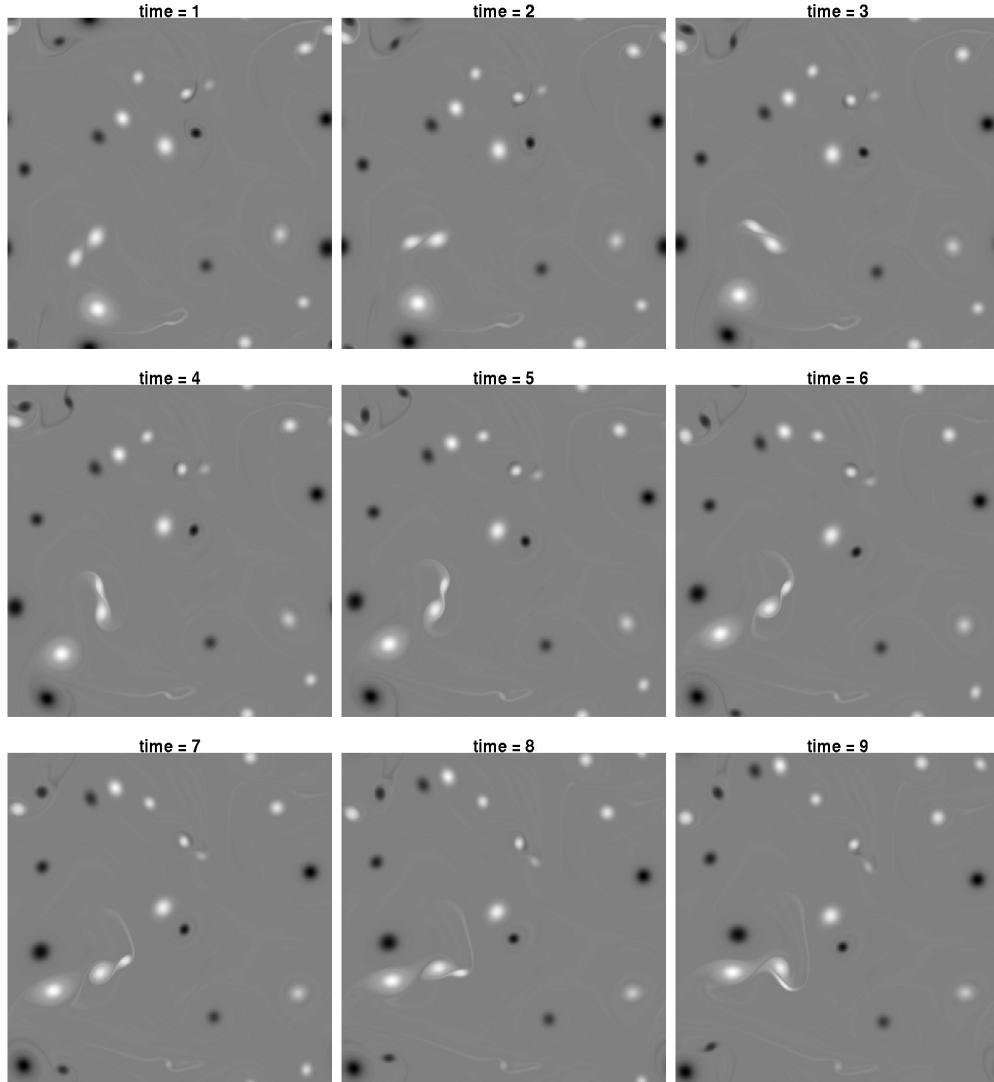


Figure 2: 2D Turbulence Simulation

3 A Stochastic Model for Target Tracking

In this section we propose a stochastic model for solving the above multiple target tracking problems. Throughout the entire modeling process, we aim to achieve the following two important and also somewhat conflicting goals: (i) we want to incorporate as much as possible of our physical understanding of the scientific problems into the model, and (ii) for computational feasibility, we want the resulting likelihood function to be quickly and accurately evaluated.

Define a path, $(X(t), Y(t))$, as the coordinates of the centroid of a target at time t . We assume that the image sequence is sampled from a continuous process at discrete times $\mathbf{t} = (t_1, t_2, \dots, t_n)$.

We shall focus on 2D settings, but our approach can be easily generalized for higher dimensional problems. We will model the path of any target by a 2D random process. This is complicated however by the occurrences of birth, death, splitting and merging of targets along with missing observations and false alarms.

The proposed model consists of the following five sub-models: (i) the *Event Model* that controls the birth, death, merging and splitting of targets; (ii) the *Observability Model* that determines when a real target is detected or missing; (iii) the *Location Model* that describes the movements of the targets; (iv) the *Attribute Model* that incorporates other characteristics of the targets (e.g., size, orientation, or intensity); and (v) the *False Alarm Model* that handles the occurrence of false alarms.

3.1 Target Event Model

The Event Model is a continuous time Markov model that determines how and when birth, death, merging or splitting events occur. The Markov assumption implies that the times between successive events are independent exponentially distributed random variables. While the best distribution for modeling event waiting times is problem dependent, the memoryless property of the exponential distribution does seem to be a realistic assumption for the the storm and vortex problems discussed above.

In the Event Model the rates at which birth, death, splitting and merging events happen are given by λ_b , $N(t)\lambda_d$, $N(t)\lambda_s$, and $(N(t) - 1)\lambda_m$ respectively, where $N(t)$ is the number of targets in existence at time t . It is assumed that the initial number of targets N_0 follows a Poisson distribution: $N_0 = N(t_1) \sim \text{Poisson}(\lambda_0)$. Notice that the rates of death and splitting events are proportional to the number of targets in existence. This is because every target has an individual rate of dying (or splitting) which is independent of other targets. For merging events, it may seem more natural for the rate to be proportional to $\binom{N(t)}{2}$, the number of pairs of targets. However, we adopted $(N(t) - 1)\lambda_m$ for the following two reasons. First, if we used $\binom{N(t)}{2}$, the problem would not be scale invariant in the following sense. If $N(t)$ is scaled up by a factor of two, then $\binom{N(t)}{2}$ will be scaled up by a factor of 4 instead of 2. Second, it is also intuitive to use $(N(t) - 1)\lambda_m$ as the

rate. This is because each target should only be merged with its closest neighbor, and there are at most $N(t) - 1$ such closest pairs to consider when there are $N(t)$ targets.

The following notation will be used to describe the Event Model

$$\begin{aligned}
U_{b,j} &= \text{number of births in the interval } [t_j, t_{j+1}) \\
U_{d,j} &= \text{number of deaths in the interval } [t_j, t_{j+1}) \\
U_{s,j} &= \text{number of splits in the interval } [t_j, t_{j+1}) \\
U_{m,j} &= \text{number of mergers in the interval } [t_j, t_{j+1}).
\end{aligned} \tag{1}$$

We will write $\mathbf{U}_b = (U_{b,1}, \dots, U_{b,n})$ and similarly for \mathbf{U}_d , \mathbf{U}_s , and \mathbf{U}_m . Also, denote the collection of N_0 and the \mathbf{U} 's by $\mathcal{U} = (N_0, \mathbf{U}_b, \mathbf{U}_d, \mathbf{U}_s, \mathbf{U}_m)$.

Each target, regardless of its status (e.g., alive or dead), will be uniquely identified by a positive integer starting from 1. We shall call such integers *indices*. The initial targets alive at time t_1 are arbitrarily labeled with indices 1 through N_0 . The following actions will be taken at the time whenever any one of the four possible events happens. When there is a birth the new target will be given the next available index. For example, if there are already 10 targets in the model (some currently alive, some could be dead), these targets would have been labeled uniquely with indices from 1 to 10, and the new target will be given an index of 11. When there is a death, all targets that are still alive are equally likely to be selected as the one that dies. When there is a split, all of the living targets are equally likely to be the parent, and the children will be given the next two available indices. Finally, for merging events all of the possible pairs of all living targets are equally likely to be the parents, and the child will be given the next available index.

Notice that the assumptions that all pairs of events are equally likely appears to be in contradiction to the principle that only close targets are eligible to merge together. We will rectify this issue in the Location Model to be described in Section 3.3. In short, locations of the parents of a merger are conditioned to be “close” to each other right before the merger. This shifts the burden of enforcing the property that “only close targets merge together” to the location model. This leads to an important simplification of the likelihood calculation since the location model depends

on the Event Model but not vise-versa.

We will specify which targets were involved in the events by

$$\begin{aligned}
V_{b,j} &= \text{the collection of indices of targets that were born in the interval } [t_j, t_{j+1}) \\
V_{d,j} &= \text{the collection of indices of targets that died in the interval } [t_j, t_{j+1}) \\
V_{s,j} &= \text{the collection of triplets } (i_1, i_2, i_3) \text{ where } i_1 \text{ is the index of the parent and} \\
&\quad i_2, i_3 \text{ are the children for every split in the interval } [t_j, t_{j+1}) \\
V_{m,j} &= \text{the collection of triplets } (i_1, i_2, i_3) \text{ where } i_1, i_2 \text{ are the indices of the parents and} \\
&\quad i_3 \text{ is the child for every merger in the interval } [t_j, t_{j+1}). \tag{2}
\end{aligned}$$

Let $\mathbf{V}_b = (V_{b,1}, \dots, V_{b,n})$ and similarly for \mathbf{V}_d , \mathbf{V}_s , and \mathbf{V}_m . The collection of all the \mathbf{V} 's will be denoted as $\mathcal{V} = (\mathbf{V}_b, \mathbf{V}_d, \mathbf{V}_s, \mathbf{V}_m)$.

Lastly, it should be noted that this is a hidden Markov model in that we do not actually observe the variables \mathcal{U} , and \mathcal{V} from the data. Predicting these variables is part of the tracking problem. This will be described further in Section 4.

3.2 Observability Model

Now we discuss our approach to modeling missing targets, that is, real targets that exist but were not detected in some image frames. It is certainly not ideal to simply allow these targets to die when they have escaped from detection and start new paths when they appear again. Instead, for such a missing target, a good tracking method should be able to impute its existence and return one single path as its track estimate. In the tracking literature missing targets are usually modeled using *iid* Bernoulli random variables. Here we will adopt this approach as well. That is, at any time t , if a target exists, it has probability P_d of being detected and producing an observation. This is also assumed independent over time.

3.3 Target Location Model

When a target is determined to exist by the Event Model, we model the path $(X_i(t), Y_i(t))$ of the i^{th} target with a Gaussian process, which is commonly used in many tracking applications. We also assume the target paths are independent of other targets unless they are required to split or merge as determined by the Event Model. The dependency introduced by splitting and merging will be described later, and first we present the distribution of $X_i(t)$ under generic conditions. The distribution of $Y_i(t)$ will be similar with the obvious changes in notation and parameters and independent of $X_i(t)$.

Let the x component of location and velocity of the i^{th} target at time t be denoted by $X_i(t)$ and $X'_i(t)$ respectively. Also denote the time of initiation of the i^{th} target by ξ_i . If the i^{th} target exists at the first observation time t_1 , then it is assumed that $\xi_i = t_1$. Then

$$X_i(t) = X_i(\xi_i) + X'_i(\xi_i)(t - \xi_i) + \sigma_i G_i(t - \xi_i) \quad (3)$$

where $G_i(t)$ is some continuous mean zero Gaussian process, for which we have chosen to use an integrated Brownian motion (IBM). This is sometimes referred to as a nearly constant velocity model and is popular for target location within the tracking community. In addition, a comparison of IBM path realizations to those produced by storms and vortexes revealed that the IBM model was flexible enough to represent these paths well.

The initial position $X_i(\xi_i)$ and the initial velocity $X'_i(\xi_i)$ depend on whether the target resulted from a birth, merging or splitting event. More detail for each of these cases is given below.

Initial Conditions for a Target Resulting from a Birth Event. Suppose that the i^{th} target is the result of a birth. It is assumed that the initial position and velocity are Gaussian, $X_i(\xi_i) \sim \mathcal{N}(\mu_{X_0}, \sigma_{X_0}^2)$ and $X'_i(\xi_i) \sim \mathcal{N}(\mu_{X'_0}, \sigma_{X'_0}^2)$. For the two scientific problems described above, it may also seem reasonable to use a uniform distribution to model the initial location $X_i(\xi_i)$. However, very often the likelihood of a uniform distribution can be satisfactorily mimicked by sufficiently increasing the variance of a normal distribution. Thus we will keep the original Gaussian assumption for mathematical convenience.

Initial Conditions for a Target Resulting from a Merging Event. Now suppose that the i^{th} target is the child resulting from a merging event. Let $p_i = (p_{i,1}, p_{i,2})$ be the vector containing the indices of the two parents. Suppose for now that size information is available and it is realistic to model the expected size as a constant as in Section 3.4. Use $E(S_j)$ to denote the mean size of the j^{th} target. Physically, the initial position or centroid of the child should be the average, weighted by size, of the positions of the parents at the time of merger,

$$X_i(\xi_i) = \left(\frac{E(S_{p_{i,1}})}{E(S_i)} X_{p_{i,1}}(\xi_i) + \frac{E(S_{p_{i,2}})}{E(S_i)} X_{p_{i,2}}(\xi_i) \right).$$

Also, by conservation of momentum, the initial velocity of the child should be the weighted average of the velocities of the parents at the time of merger,

$$X'_i(\xi_i) = \left(\frac{E(S_{p_{i,1}})}{E(S_i)} X'_{p_{i,1}}(\xi_i) + \frac{E(S_{p_{i,2}})}{E(S_i)} X'_{p_{i,2}}(\xi_i) \right).$$

If there is no size information available then we can let the initial position of the child be the simple average of the positions of the parents, plus perhaps a small amount of noise $\psi_{m,i}$. Figure 3 displays a physical representation of this. We can also mimic the conservation of momentum by taking the child's velocity to be the simple average of the parent velocities plus noise. This yields

$$X_i(\xi_i) = \frac{1}{2} (X_{p_{i,1}}(\xi_i) + X_{p_{i,2}}(\xi_i)) + \psi_{m,i} \quad (4)$$

$$X'_i(\xi_i) = \frac{1}{2} (X'_{p_{i,1}}(\xi_i) + X'_{p_{i,2}}(\xi_i)) + \psi'_{m,i} \quad (5)$$

where $\psi_{m,i} \sim \mathcal{N}(0, \sigma_{X_m}^2)$ and $\psi'_{m,i} \sim \mathcal{N}(0, \sigma_{X'_m}^2)$. Presumably, $\sigma_{X_m}^2$ and $\sigma_{X'_m}^2$ are small so that the new target location and velocity are likely to be close to the averages of the parents.

Parent Locations at the Time of a Merging Event. Notice that in our modeling so far, the two parent targets are not required to be close to each other at the time of a merging event. To ensure that the parents move close to each other before merging, the difference between locations of the parents at the time of merger is conditioned to be small. This is done as follows.

Let $d = (d_1, d_2, d_3)$ be a vector containing the indices of the three targets involved in a merging

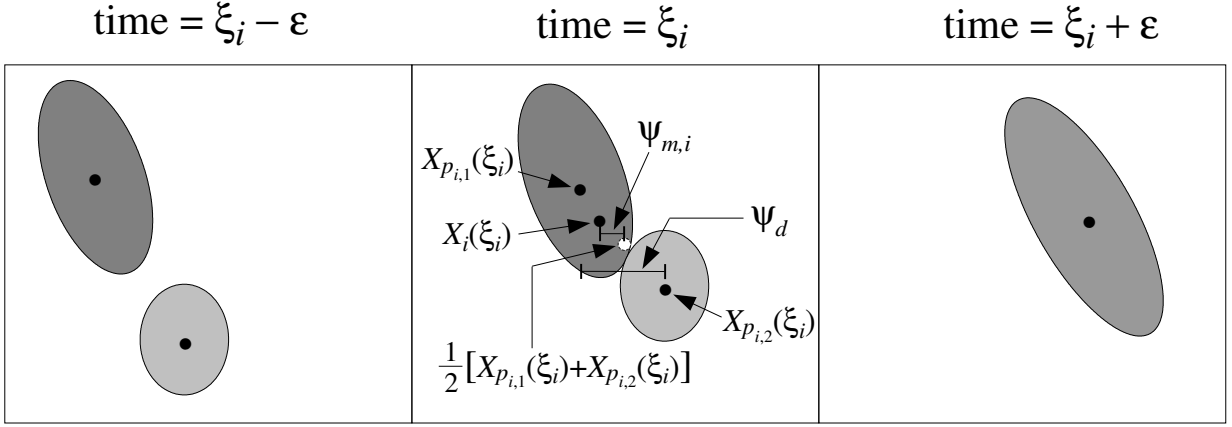


Figure 3: Physical Description of a Merger

event where d_1 and d_2 are the parents while d_3 is the index of the child. Let D be the difference in location between the two parents at the time of merger plus a noise term,

$$D = X_{d_1}(\xi_{d_3}) - X_{d_2}(\xi_{d_3}) + \psi_d \quad (6)$$

where $\psi_d \sim \mathcal{N}(0, \sigma_{X_d}^2)$ and independent of the targets. If σ_{X_d} is small, then it is likely that ψ_d is small in absolute value. If we then condition the model for X_{d_1} and X_{d_2} on the event $D = 0$, this will ensure that the parents are only a small distance ψ_d apart at the time of the merging event. In Figure 3 once again, we see a merging event with a possible realization of ψ_d .

In general, there will be $N_m = \sum_{j=1}^n U_{m,j}$ merging events during the time window $[t_1, t_n]$. We will condition the target paths on all of these mergers in a manner similar to that above. This is described more precisely as follows. Let D_i be the D from (6) and $\psi_{d,i}$ be the corresponding ψ_d for the i^{th} merging event, $i = 1, \dots, N_m$. We then condition the model for (X_1, \dots, X_M) on the event $\{(D_1, \dots, D_{N_m}) = (0, \dots, 0)\}$, where M is the total number of targets that existed before time t_n .

Here we remark that our approach for handling merging, as illustrated by Figure 3, is a very realistic physical description of merging targets for many real problems, including the two scientific problems described earlier.

Initial Conditions for a Target Resulting from a Splitting Event. Suppose that the i^{th}

target is initiated by a splitting event. That is, this i^{th} target is one of the two children resulting from a splitting event. To keep notation consistent, let $p_{i,1}$ be the index of the parent. The location of a child resulting from a split is modeled by

$$X_i(t) = X_{p_{i,1}}(\xi_i) + \psi_{s,i} + \left[X'_{p_{i,1}}(\xi_i) + \psi'_{s,i} \right] (t - \xi_i) + \sigma_i G_i(t - \xi_i) \quad (7)$$

where $\psi_{s,i} \sim \mathcal{N}(0, \sigma_{X_s}^2)$ and $\psi'_{s,i} \sim \mathcal{N}(0, \sigma_{X'_s}^2)$. Similar to that for merging events, the initial position and velocity of a new child from a splitting event is the same as that of the parent plus a random error. It is assumed that $\sigma_{X_s}^2$ is small so that the new targets are likely to appear close to where the parent split. Similarly, $\sigma_{X'_s}^2$ should be small so that the new targets have a velocity similar to that of their parent.

If size information is available then the conservation of momentum assumption can also be imposed on the two new paths from the children splitting off from the parent. This can be achieved by conditioning in a similar manner as for parent locations at the time of a merging event.

Measurement Error. Due to imperfect detection or some other reasons, we do not always observe the exact locations of the targets. Rather, instead of $X_i(t)$, we observe a noisy version of it. We shall assume that the measurement errors are additive, Gaussian and independent over time. That is, for $j = 1, \dots, n$, we observe

$$X_i^*(t_j) = X_i(t_j) + \varepsilon_{i,j} \quad \text{with} \quad \varepsilon_{i,j} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_{X_e}^2).$$

3.4 Target Attribute Models

In this section we will describe several models that can be applied in conjunction with the Event and Location Models to improve the tracking results when some auxiliary information about target attributes is available. We shall present a few special cases of attributes that are commonly available: size, orientation and intensity. Other attributes may be handled in a similar manner.

Size. In the 2D case, the size of a target is the area and, similar to the approach in Angelova

& Mihaylova (2006), we will use the length of the minor and major axes, $R_1(t)$ and $R_2(t)$, of the best fitting ellipse of the target to characterize it (see Figure 4). We will assume that a best fitting ellipse for each target has already been obtained; e.g., by standard imaging techniques as described in Rosenfeld & Kak (1982).

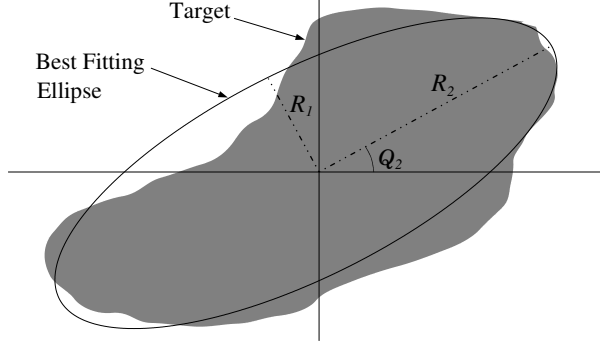


Figure 4: R_1 , R_2 , and Q_2

Let $R_{1,i}(t)$ and $R_{2,i}(t)$ be the minor and major radii respectively for the i^{th} target at time t . These two quantities $R_{1,i}(t)$ and $R_{2,i}(t)$ will be modeled as log-normal random variables with parameters $(\mu_{R_{1,i}}, \sigma_{R_{1,i}}^2)$ and $(\mu_{R_{2,i}}, \sigma_{R_{2,i}}^2)$ respectively. The log-normal distribution is commonly used to describe the distribution of lengths (Wang & Lee 2006). These observations are also assumed to be independent over time. One could certainly make other more realistic assumptions, such as allowing the size of a target to change as a positive continuous process. However, this assumption leads to complications in the likelihood calculation when there are splitting and merging events. This in turn drastically increases the computational load so we will not consider this option further.

Intensity. For many tracking problems, the intensity $I(t)$ of a target can be defined in various meaningful manners. For example, for the storm application, it can be defined as the maximum rainfall rate, the average of rainfall rates, or even some combination of the two. Such intensities can be modeled in a manner similar to size. We can assume some appropriate random process for the observations. For the storm application in Section 6, the intensity did not seem to change much over time, and we found that treating the observations as *iid* from a lognormal distribution was adequate.

Orientation. The orientation of a target can be measured by the angle, Q_2 of the major axis corresponding to R_2 as shown in Figure 4. We assume that Q_2 follows a von Mises distribution on $[0, \pi]$ with parameters α_i and β_i (Fisher 1995). As with intensity and radii, we assume that $Q_2(t)$ is *iid* over time in the storm application of Section 6.

3.5 False Alarm Model

The modeling of false alarms, also known as clutter, is divided into three parts (event, location and attributes) in a manner similar to that for targets.

False Alarm Event and Location. In most existing tracking methods, the number of false alarms in each image is assumed to be Poisson while the locations are typically assumed uniform throughout the image. This is equivalent to a homogeneous spatial Poisson process (Diggle 2003). We will also assume that the distribution of false alarms is a spatial Poisson process (possibly heterogeneous) that is independent over time with intensity function $\rho(x, y)$. In addition, we recommend using $\rho(x, y) = \lambda_f[X_i(\xi)](x)[Y_i(\xi)](y)$, where $[X_i(\xi)]$ is the density of the initial x -location of a target resulting from a birth, and similarly for $[Y_i(\xi)]$ with respect to the y -location. This is to ensure that the contribution of the initial location to the likelihood will not influence the probability that an observation is a target versus a false alarm; see Section 4. Notice that λ_f is then the expected number of false alarms at each time point. Unlike the case for real targets, there is no need for an Observability Model for false alarms, as “missing false alarms” will never exist.

False Alarm Attributes. If any attributes are used to model the real targets, they must also be used to model the false alarms. This is to guarantee that the likelihood computed by assuming a group of sequential observations originated from a real target would be comparable to the likelihood computed by assuming the same observations were false alarms. We also propose that a false alarm attribute have independent and identical distribution for each false alarm occurrence. The conceptual reason for the identical distribution is that all of these false observations are artifacts of the same device. Hence it is reasonable to assume the parameters are the same.

4 The Tracking Estimate

In this section we formally define the estimand of our tracking problem and propose a method for estimating it. The setup for this problem is as follows. We collect data at the following times t_1, \dots, t_n . At the j^{th} time step m_j observations are detected from the j^{th} image frame. Each of these m_j observations is either a real target or a false alarm. Let $Z_i(t_j)$ be the i^{th} observation at time t_j , $i = 1, \dots, m_j$. Each $Z_i(t_j)$ is a vector of the location values for either a target or a false alarm. Depending on the problem, it may also include the values of attribute variables.

Let $\mathcal{Z} = \{Z_i(t_j) : j = 1, \dots, n; i = 1, \dots, m_j\}$ be the collection of observations at all times. From our data, \mathcal{Z} , we need to decide whether each observation, $Z_i(t_j)$, was originated from a real target or a false alarm. In addition, if it was from a real target, we also need to decide which target it should be assigned to. Note that each observation can be assigned to only one target and each target can only have one observation assigned to it. We will create the variable $p_i(t_j)$ to be the index of the target that observation $Z_i(t_j)$ was originated from. We can define the index for a false alarm to be 0.

Write $\mathcal{P} = \{p_i(t_j) : j = 1, \dots, n; i = 1, \dots, m_j\}$. So for a given \mathcal{Z} , \mathcal{P} will specify the tracks of each target. On top of that we must also specify the events (births, deaths, splitting and merging) that occurred with the variables \mathcal{U} and \mathcal{V} defined in (1) and (2) respectively. The variables \mathcal{U} and \mathcal{V} together with \mathcal{P} will denote a solution to the tracking problem. Let our estimate of the tracking solution $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ be denoted $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$. Notice that in the usual statistical terminology this is actually called a prediction problem, since $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ is a random variable in our framework. We chose to call it the *tracking estimate* instead of the tracking prediction to avoid the potential confusion with track prediction (of future target locations).

4.1 Calculating $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$

First we delay the issue of parameter estimation to Section 4.4, and assume for the moment that all the parameters in the model, such as $\lambda_0, \lambda_b, \dots, P_d, \sigma_i$'s etc., described in Section 3 are known; see the first paragraph in Section S.6 for a complete listing of the model parameters. We will use the notation $[X]$ to denote the probability density function of the random variable X , $[X](x)$ to

denote $[X]$ evaluated at x and $[X | Y]$ to denote the conditional density of X given Y .

To achieve our tracking estimate, we compute the conditional density of $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ given \mathcal{Z} ,

$$[\mathcal{U}, \mathcal{V}, \mathcal{P} | \mathcal{Z} = z](u, v, p). \quad (8)$$

Note that this is a probability mass function since the variables $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ are discrete. This can be computed fairly efficiently because the likelihood of our model conveniently factorizes into several conditional densities. See Section S.3 for details.

From this it is natural to define our tracking estimate as

$$(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}}) = \arg \max_{u, v, p} [\mathcal{U}, \mathcal{V}, \mathcal{P} | \mathcal{Z} = z](u, v, p). \quad (9)$$

Even more, we can calculate the probability that $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ is the correct solution given the data \mathcal{Z} as $[\mathcal{U}, \mathcal{V}, \mathcal{P} | \mathcal{Z} = z](\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$.

A common but major difficulty in most problems is that it is not computationally feasible to enumerate all possible tracking solutions and calculate their likelihood values. To overcome this issue, we have developed a variant of the Multiple Hypothesis Tracking (MHT) algorithm of Reid (1979), the details of which are described in Section 4.3.

Our MHT algorithm will locate an approximation to the solution that maximizes (8), and hence it will provide an point estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ to our tracking problem. Furthermore, by providing an estimate of the density function (8), this MHT algorithm will also allow us to make various probability statements about the solution. This is achieved as follows. Upon completion the MHT algorithm provides approximately the K best solutions; i.e., those with highest likelihood values. We will discuss the choice of K in the next subsection. Label these K solutions (u_i, v_i, p_i) for $i = 1, \dots, K$ and let \mathcal{K} denote the set of these K solutions. If we assume that the correct solution is in \mathcal{K} , then we can calculate the conditional density of $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ given $\mathcal{Z} = z$ and the event $(\mathcal{U}, \mathcal{V}, \mathcal{P}) \in \mathcal{K}$,

$$[\mathcal{U}, \mathcal{V}, \mathcal{P} | \mathcal{Z}, (\mathcal{U}, \mathcal{V}, \mathcal{P}) \in \mathcal{K}](u, v, p | z). \quad (10)$$

In practice we can then compute the distribution given in (10) to approximate that in (8).

4.2 Confidence Sets

Equation (8) provides the distribution of all possible tracking solutions given the data \mathcal{Z} , which is much more informative than just the estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$. For example, by summing over the relevant probabilities, we could calculate the probability that a given observation is a target or a false alarm, or the probability that two targets merged, etc. Another innovative use of (8) is confidence set construction. That is, we could construct a confidence set C in the sense that the probability that C contains the correct solution is $\geq (1 - \alpha)100\%$. A major challenge for this is how to summarize and display all elements (i.e., different solutions of paths) in C in a meaningful and informative manner. We will report our work on this topic in a future paper.

4.3 A Modified MHT Algorithm

We have modified the MHT algorithm of Reid (1979) to approximate our tracking solution (9). Major steps of this modification are described next while a more detailed description is given in Section S.4. However, we shall begin with a brief description of the original MHT algorithm of Reid (1979).

First we note that in the tracking literature a possible solution to a tracking problem is also called a *hypothesis*, and hence the name of the algorithm. The original MHT algorithm assumes that an objective function (e.g., likelihood) has been constructed for evaluating the score of any solution at any time step. It processes the data sequentially. At each time step t it keeps a set of the K hypotheses with highest likelihood, where K is pre-specified by the user. Then at the next time step $t + 1$ it forms the collection of all feasible hypotheses given the new data for each of the hypotheses from the previous time t . It then prunes the list of the new hypotheses back to K of them, by eliminating those that have low scores. The algorithm continues in a similar fashion until the data sequence finishes. At the end of the algorithm K sets of tracking estimates will be obtained and the one that corresponds to the best score value will be taken as the final tracking estimate.

Here is an overview of our modified MHT algorithm. For all of the details including “gating” procedures and other computational efficient shortcuts see Section S.4. At time t_1 , the beginning of

the image sequence, we consider all combinations of each observation either treated as a real target or a false alarm. Thus if there are m_1 initial observations, then there are 2^{m_1} possible combinations. Now imagine that we have a set of solutions for the observations through time t_{j-1} . We then take the new observations \mathbf{Z}_j at time t_j to form updated solutions based on all possible combinations of the following possibilities: we assume that each observation $Z_i(t_j) \in \mathbf{Z}_j$ is either (i) an observation from an existing target track, (ii) the first observation from a target resulting from birth, (iii) the first observation from a target resulting from split, (iv) the first observation from a target resulting from merger, or (v) a false alarm. We also assume that any existing track that does not receive a new observation must either (i) become (or stay) missing or (ii) terminate.

To reduce computational load, we only keep a subset of the new solutions (those with the highest likelihood) to form solutions for the next time step t_{j+1} . The actual number of solutions K that we keep through to the next time step is determined by the following. Let L_j be the likelihood value of the best solution at time t_j . At each time step t_j , we keep only solutions that have likelihood values greater than cL_j , where $c < 1$ is a user-defined parameter. In addition we limit the total number of hypotheses kept to be less than K_s . In our implementation we set $c = e^{-10}$ and $K_s = 200$.

4.4 Incorporating Estimation of Static Model Parameters

Up until this point case we have assumed the values of all static model parameters to be known. This is the approach that is taken by most tracking methods. Here we consider the more realistic case that these parameter values are unknown. First we collect all the static model parameters such as λ_0 , λ_b , σ_i^2 , etc. into the vector θ and write the density in (8) as $[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}]_\theta$ to explicitly represent the dependence on θ .

To estimate θ , one could use maximum likelihood, or any other suitable estimates; see Section S.5. Since these estimates depend on the variables $(\mathcal{U}, \mathcal{V}, \mathcal{P})$, we could allow each of the solutions (u, v, p) that we consider in the above modified MHT algorithm to have its own parameter estimate which we denote $\hat{\theta}(u, v, p)$. Notice that this will lead to an overly optimistic likelihood value

$$[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}]_{\theta=\hat{\theta}(u,v,p)}(u, v, p \mid z) \quad (11)$$

for each of the solutions. However, based on the work of Storlie et al. (2007), it can be shown under certain conditions that as the time between observations tends to zero,

$$(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}}) = \arg \max_{u,v,p} [\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}]_{\theta=\hat{\theta}(u,v,p)}(u, v, p \mid z) \quad (12)$$

is equal to $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ eventually almost surely. Hence we use the distribution defined in (11) to calculate likelihood values within the MHT iterations and to decide which solutions that we shall keep for propagation to the next time step.

Although the function in (11) is useful for calculating the point estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$, it is not a good approximation of the distribution in (8). This is because each argument, (u, v, p) , is given its own estimated value for θ which introduces a bias that remains even asymptotically. In order to estimate (8), we can however use $[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}]_{\theta=\tilde{\theta}}$ where $\tilde{\theta}$ is a consistent estimator of θ . This will ensure that the probabilities given by $[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}]_{\theta=\tilde{\theta}}$ have a frequentist interpretation, at least in an asymptotic sense. A natural candidate for $\tilde{\theta}$ is given by $\tilde{\theta} = \hat{\theta}(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$. As long as the estimator $\hat{\theta}(\mathcal{U}, \mathcal{V}, \mathcal{P})$ is consistent, then $\hat{\theta}(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ is also consistent by the result given in Storlie et al. (2007) mentioned above.

Thus, if the MHT algorithm finishes with K solutions $\{(u_i, v_i, p_i)\}_{i=1}^K$ at the last step, we could use

$$\left\{ [\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}, (\mathcal{U}, \mathcal{V}, \mathcal{P}) \in \mathcal{K}]_{\theta=\hat{\theta}(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})}(u_i, v_i, p_i \mid z) \right\}_{i=1}^K \quad (13)$$

as an estimate of (8) to allow us to make various probability statements concerning our final tracking estimates. We used (13) to compute the probabilities to be reported in Figure 5 and Section S.6.

Lastly we remark that in practice the researcher can, based on physical consideration on the tracking problem at hand, impose various limits on the parameter estimates. This will serve to limit the amount of bias introduced into the approximation given in (13). For example, the researcher is usually familiar with the range for the number of targets and/or false alarms so that effective limits can be imposed for the corresponding model parameters. This is certainly an improvement over the need for specifying these parameter values exactly.

4.5 Summary

Here we summarize the major steps of the proposed tracking method:

1. To initialize the algorithm consider all combinations of each observation from \mathbf{Z}_1 either treated as a real target or a false alarm. Denote the set of these combinations as \mathcal{K}_1 . Set $t = 2$.
2. With \mathbf{Z}_t and \mathcal{K}_{t-1} , obtain the set of all possible solutions for time $\{1, \dots, t\}$ by enumerating all the possibilities listed in the third paragraph of Section 4.3.
3. Calculate the likelihood values for all these solutions using (11). Keep the K solutions that have the K largest likelihood values, where K is determined by the method described in the last paragraph of Section 4.3. Collect these K best solutions into \mathcal{K}_t .
4. Increment t and repeat Steps 2 and 3 until all \mathbf{Z}_t are processed. Denote the last set of tracking solutions as \mathcal{K}_n .
5. Take the solution from \mathcal{K}_n with the highest likelihood value as the final tracking estimate. In addition, using (13) one could use all members of \mathcal{K}_n to provide an estimate for the distribution (8).

5 Simulated Data Results

In this section, we present the results of the above proposed tracking method on a simulated data set. For a full description and discussion of more extensive simulation results, see Section S.6 of the online supplementary document.

In this example, the data \mathcal{Z} are assumed to come from the model described in Section 3. The random motion component $G_i(t)$ is an integrated Brownian Motion for all targets. The event parameters are set at $\lambda_0 = 4$, $\lambda_b = 0.1$, $\lambda_d = 0.02$, $\lambda_s = 0.06$, $\lambda_m = 0.08$ and $\lambda_f = 8.0$. All parameter values were chosen to mimic the rainfall data of Section 6. Values for other model parameters along with the limits imposed on the parameters for the purposes of estimation are also given in Section S.6.

A realization from this model is shown in Figure 5. All the observations from all time steps are plotted together in one plot. Observations from time t_j are labeled ' t_j '. The correct solution and the top four alternative solutions are provided along with their estimated probabilities approximated by (10). In this example we would be about 90% confident that our point tracking estimate is the correct solution. In fact, our estimate is the correct solution in this case.

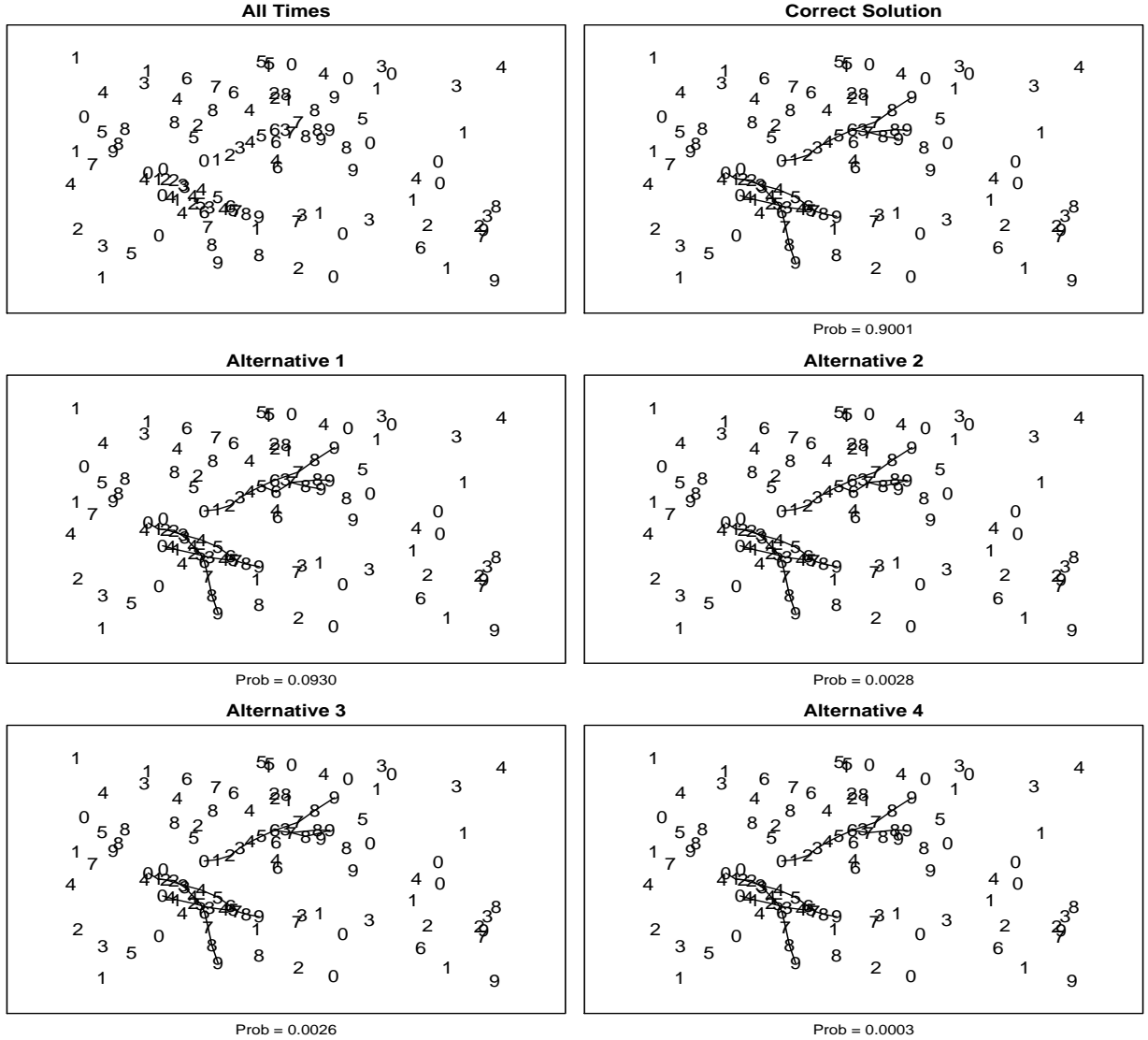


Figure 5: Tracking estimates of the proposed method on simulated data.

6 Application to Rainfall Data

Here we consider tracking the storms that evolve during the morning of July 14, 1996 in the images of Figure 1. We will utilize available attribute data: size (radii of best fitting ellipse) and orientation. It can be concluded from the simulation results of Section S.6 that attribute information can be very beneficial for tracking estimation.

For the purposes of this problem, we are interested in tracking the mesoscale convective systems, which are usually defined to be storms with major axis longer than 100 km. This corresponds to roughly 1° latitude or longitude in Figure 1. Before we can track the storms, we must first identify them from the images and measure their location and attributes. An effective detection algorithm was used for this step and is described in Section S.7.

Recall that the images in Figure 1 are separated by 30 minutes. So this small illustration covers a time span of 2.5 hours. Figure 6 shows the same images as in Figure 1 after the initial processing of the detection algorithm. It is the ellipses in these images that we will actually track.

Figure 7 shows the results of applying the proposed tracking method to the images in Figure 6. The limits for parameter estimation were set to the same as those in the simulations of Section S.6. The estimated tracks of the storms given by the tracking algorithm are displayed in black. Whenever there is a merging event an orange line is drawn connecting the parents to the child. Whenever there is a splitting event a magenta line is drawn connecting the parent to the children.

Notice that the estimate recovers the merging event that occurred between 1:00am and 1:30am, where the two storms in the south west corner of South Dakota merged together into one storm. Also in this same time interval, the large system over Alabama and the panhandle of Florida split into two smaller systems. The reader is also referred to the same website that contains the online supplementary document (end of Section 1) to see a video of the raw data, the processed data, and the tracking solution with the path lines. The corresponding links from the above webpage are *Rain Fall Video*, *Cleaned Rainfall Images*, and *Paths Given by Tracking Algorithm*. These videos cover a longer time span as well, from 1:00am to 1:00pm on July 14, 1996.

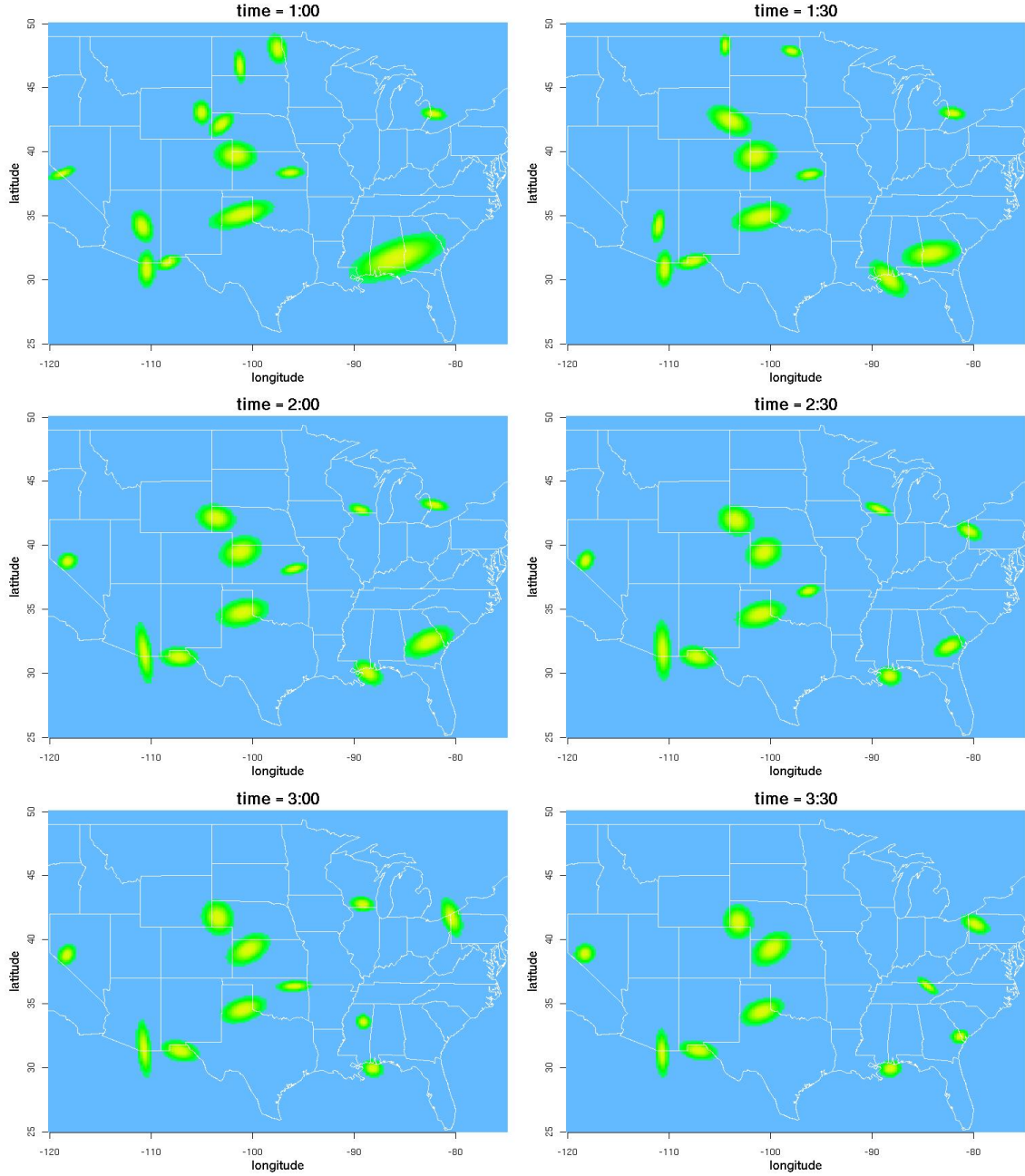


Figure 6: Best Fitting Ellipses of Radar Reflectivity Images

7 Application to 2D Turbulence

Here we consider tracking the vortexes in the 2D turbulence simulation displayed in Figure 2. We will again utilize the size information but forgo the use of orientation, since all of the vortexes are nearly circular. The detection algorithm used to identify the vortexes is similar to that used for the storms in Section 6.

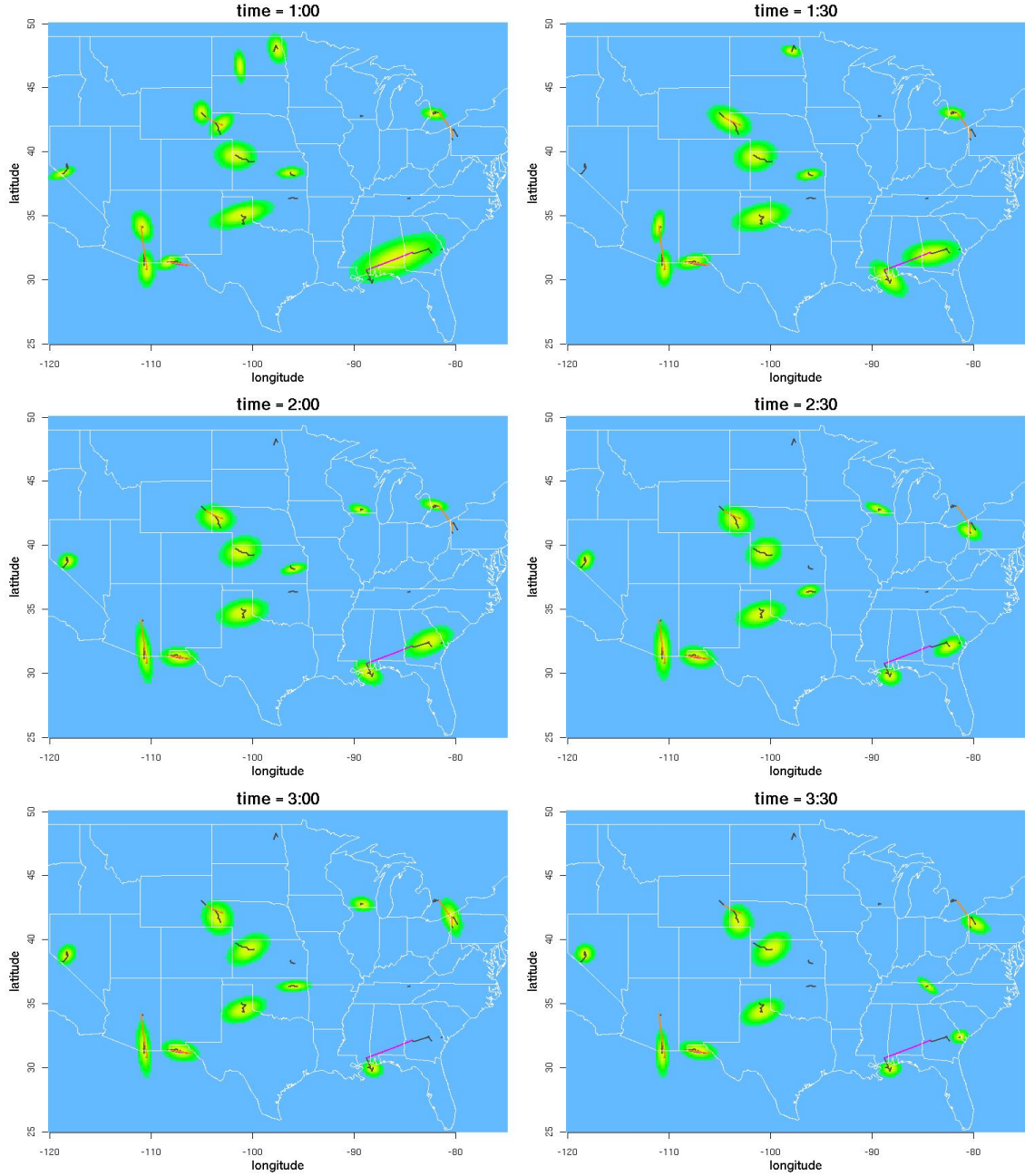


Figure 7: Best Fitting Ellipses with Estimated Paths Superimposed

Figure 8 shows the resulting $(\hat{u}, \hat{v}, \hat{p})$ after applying the tracking algorithm to the processed data of Figure 2. Recall that the black vortices are spinning in a counterclockwise direction while the white vortices are spinning in a clockwise direction. The red lines indicate the paths of black vortices, while blue lines indicate the paths of white vortices. Merging events are indicated by orange lines for black vortices and green lines for white vortices. There is no splitting of targets

in this problem.

A good example of a merging event can be found between frames 8 and 9. In addition, the tracking algorithm was able to capture the fact that one of these vortexes was missing (not found by the detection algorithm) for two frames just prior to the merging event. This is illustrated by a continuous path line where the vortex disappears for two frames.

The same website given at the end of Section 1 above also contains the corresponding video of the raw data given by the link *2D Turbulence Video*. Videos of the best estimate path lines superimposed on the cleaned images as well as on the raw images are given by the links *Vortex Paths on Cleaned Images* and *Vortex Paths on Raw Images* respectively.

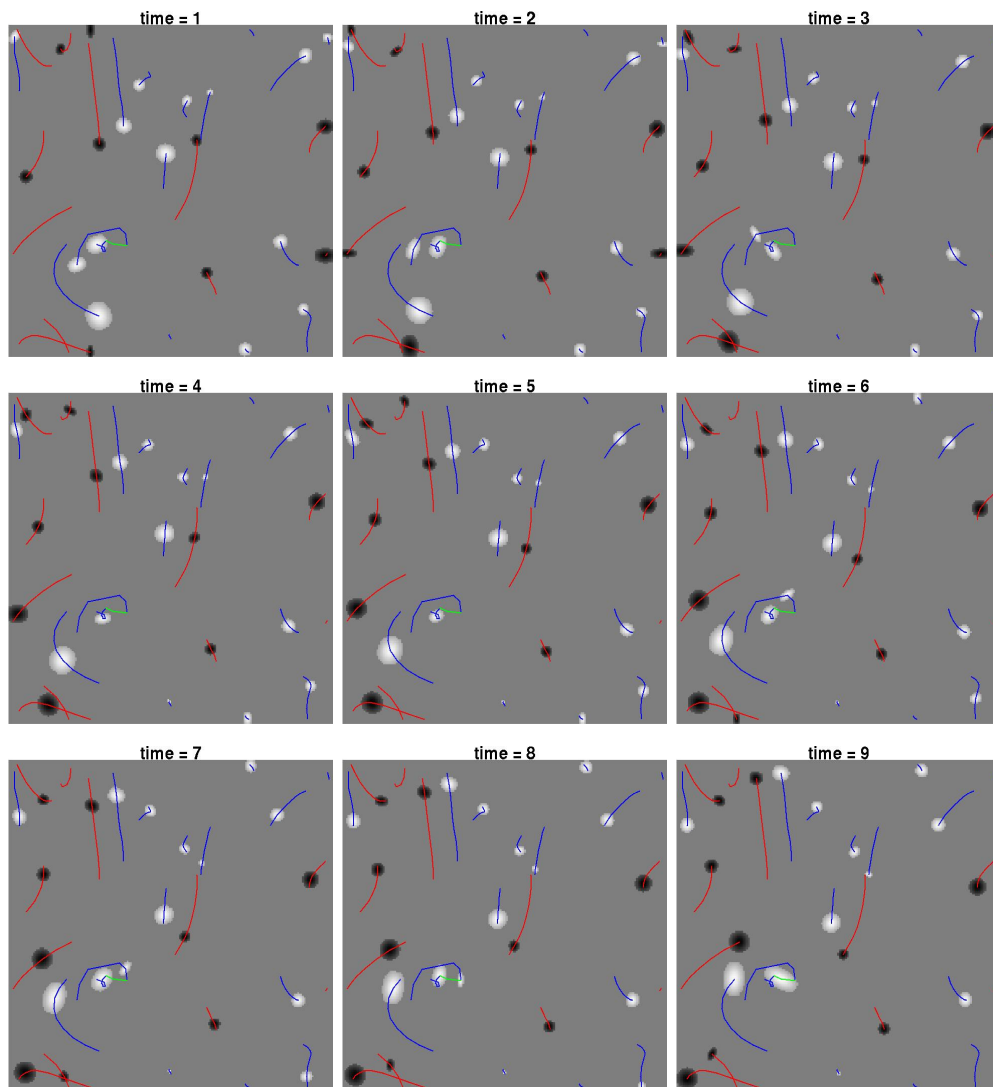


Figure 8: 2D Turbulence Simulation

8 Conclusions & Further Work

In this article we have presented an introduction to multiple target tracking. We have also developed a new stochastic model that incorporates the events of birth, death, splitting and merging of targets into the likelihood, as well as missed detections and false alarms. Track estimation is then accomplished by considering the distribution of the relevant variables given the data.

The results obtained by the simulated data provides strong evidence that this estimation procedure works very well even under the presence of false alarms and missing observations. The utility of the method was also demonstrated on two non-standard applications: radar reflectivity data collected over the United States and vortexes in 2D turbulence images. A future project is to apply the method to other more difficult tracking applications, such as smaller scale convective systems (thunder storms) and/or missile target tracking.

For multiple target tracking estimation, there remain many challenging and interesting problems that are particularly suited for statisticians to tackle. As a first example, it would be useful to develop a procedure for sampling rate determination for tracking problems. That is, to automatically determine how frequent should one collect the data.

Secondly, what would be the best approach to incorporate and utilize target attribute information when they are available? In this article we have used an ellipse to represent shape. Previously, Isard & Blake (1998) use curve segments to model target shape, allowing them to change over time, and Wang & Zhu (2004) decompose an image into basis functions, called tokens, and group these tokens together to form targets, again allowed to deform over time. However, all these methods are very much problem dependent. It would be advantageous to develop a statistically sound principle to guide us on how to incorporate attribute information into the tracking solutions.

It would also seem very beneficial to combine the detection problem with the tracking problem. This is because the inherent spatio-temporal dependency provides additional information to help the detection algorithm to predict where targets are more likely to exist in successive images. Some work has been done in this direction (e.g., Tonissen & Bar-Shalom 1998, Salmond & Birch 2001, Kreucher et al. 2005 and Boers, Driessen, Torstensson, Trieb, Karlsson & Gustafsson 2006), but the tracking models adopted tend to be relatively simple.

Lastly, virtually no rigorous attempts have been made to theoretically justify the use of any particular tracking methods. Although, as a first step, Storlie et al. (2007) investigate some theoretical properties of the estimates described in this article, there is still much more theoretical work to be done to further the understanding of the multiple target tracking problem. Statisticians are well equipped to answer these questions (especially the last one), and it is our hope that this article can stimulate further statistical investigations in various aspects of the multiple target tracking problem.

Acknowledgments

The authors thank the National Science Foundation (Grants DMS-9815344 and 0504737) for partial support of this work, particularly for the funding for the conference presentation of the preliminary version Storlie, Davis, Hoar, Lee, Nychka, Weiss & Whitcher (2004). The authors would also like to thank Tim Hoar for his help with data handling and computational resources, and Drs. Chris Davis and Jeff Weiss for their help with understanding the storm and vortex dynamics respectively. Lastly the authors are grateful to the reviewers, associate editor, and co-editors for their most constructive comments, many of which are incorporated in the current version of this article.

References

- Anderson, B. & Moore, J. (1979), *Optimal Filtering*, Englewood Cliffs, NJ: Prentice Hall.
- Andrieu, C. & Doucet, A. (2003), ‘Online Expectation-Maximization type algorithms for parameter estimation in general state space models’, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* **6**, 69–72.
- Angelova, D. & Mihaylova, L. (2006), ‘A Monte Carlo algorithm for state and parameter estimation of extended targets’, *Proceedings of the International Conference on Computational Science* **3994**, 624–631.
- Bal, B. & Alam, M. (2005), ‘Automatic target tracking in FLIR image sequences using intensity variation function and template modeling’, *IEEE Transactions on Instrumentation and Measurement* **54**, 1846–1852.
- Bar-Shalom, Y., Li, X. R. & Kirubarajan, T. (2001), *Estimation with Applications to Tracking and Navigation*, New York: John Wiley & Sons, Inc.
- Blackman, S. (2004), ‘Multiple hypothesis tracking for multiple target tracking’, *IEEE A&E Systems Magazine* **19**, 5–18.

- Blackman, S. & Popoli, R. (1999), *Design and Analysis of Modern Tracking Systems*, Boston: Artech House.
- Boers, Y., Driessen, H., Torstensson, J., Trieb, M., Karlsson, R. & Gustafsson, F. (2006), ‘Track-before-detect algorithm for tracking extended targets’, *IEE Proceedings of Radar, Sonar, and Navigation* **153**, 345–351.
- Bracco, A., McWilliams, J. C., Murante, G., Provenzale, A. & Weiss, J. B. (2000), ‘Revisiting freely decaying two-dimensional turbulence at millennial resolution’, *Physics of Fluids* **12**, 2931–2941.
- Chang, K. C. & Bar-Shalom, Y. (1984), ‘Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers’, *IEEE Transactions on Automatic Control* **29**, 585–594.
- Chen, R. & Liu, J. S. (2000), ‘Mixture Kalman filters’, *Journal of the Royal Statistical Society Series B* **62**, 493–508.
- Daum, F. E. (1994), ‘The importance of resolution in multiple target tracking’, *Signal and Data Processing of Small Targets 1994, SPIE Proceedings* **2235**, 329–338.
- Davis, C. A., Manning, K. W., Carbone, R. E., Trier, S. B. & Tuttle, J. D. (2003), ‘Coherence of warm-season continental rainfall in numerical weather prediction models’, *Monthly Weather Review* **131**, 2667–2679.
- Diggle, P. J. (2003), *Statistical Analysis of Spatial Point Patterns*, 2nd edn, New York: Edward Arnold.
- Dixon, M. (1994), *Automated Storm Identification, Tracking and Forecasting - A Radar Based Method*, PhD Thesis, University of Colorado and National Center for Atmospheric Research.
- Doucet, A., de Freitas, N. & Gordon, N. (2001), *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York.
- Doucet, A., de Freitas, N. & Gordon, N. (2003), ‘Parameter estimation in general state-space models using particle methods’, *Annals of the Institute of Statistical Mathematics* **55**, 409–422.
- Doucet, A., Godsill, S. & Andrieu, C. (2000), ‘On sequential Monte Carlo sampling methods for bayesian methods’, *Statistics and Computing* **10**, 197–208.
- Fisher, N. I. (1995), *Statistical Analysis of Circular Data*, Cambridge University Press.
- Genovesio, A. & Olivo-Marin, J. (2004), ‘Split and merge data association filter for dense multi-target tracking’, *Proceedings of the 17th International Conference on Pattern Recognition* pp. 677–680.
- Gordon, N., Salmond, D. & Smith, A. (1993), ‘Novel approach to nonlinear/non-Gaussian Bayesian state estimation’, *IEE Proceedings-F* **140**, 107–113.
- Hodges, K. I. (1994), ‘A general method for tracking analysis and its application to meteorological data’, *Monthly Weather Review* **122**, 2573–2586.
- Hodges, K. I. (1999), ‘Adaptive constraints for feature tracking’, *Monthly Weather Review* **127**, 1362–1373.

- Isard, M. & Blake, A. (1998), ‘CONDENSATION - conditional density propagation for visual tracking’, *International Journal of Computer Vision* **29**, 5–28.
- Johnson, J., Mackeen, P., Witt, A., Mitchell, E., Stumpf, G., Eilts, M. & Thomas, K. (1998), ‘The storm cell identification and tracking algorithm: An enhanced WSR-88D algorithm’, *Weather and Forecasting* **13**, 263–276.
- Julier, S. & Uhlmann, J. (2004), ‘Unscented filtering and nonlinear estimation’, *Proceedings of the IEEE* **92**, 401–422.
- Khan, Z., Balch, T. & Dellaert, F. (2005), ‘Multitarget tracking with split and merged measurements’, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* **1**, 605 – 610.
- Khan, Z., Balch, T. & Dellaert, F. (2006), ‘MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**, 1960–1972.
- Kitagawa, G. (1996), ‘Monte Carlo filter and smoother for non-Gaussian nonlinear state models’, *Journal of Computational and Graphical Statistics* **5**, 1–25.
- Koch, W. & van Keuk, G. (1997), ‘Multiple hypothesis track maintenance with possibly unresolved measurements’, *IEEE Transactions on Aerospace and Electronic Systems* **33**, 883–892.
- Kreucher, C., Kastella, K. & Hero, A. (2005), ‘Multitarget tracking using the joint multitarget probability density’, *IEEE Transactions on Aerospace and Electronic Systems* **41**, 1396–1414.
- Lakshmanan, V., Rabin, R. & DeBrunner, V. (2003), ‘Multiscale storm identification and forecast’, *Journal of Atmospheric Research* **67**, 367–380.
- Liu, J. S. & Chen, R. (1998), ‘Sequential Monte Carlo methods for dynamic systems’, *Journal of the American Statistical Association* **93**, 1032–1044.
- Mahler, R. (2003), ‘Multitarget Bayes filtering via first-order multitarget moments’, *IEEE Transactions on Aerospace and Electronic Systems* **39**, 1152–1178.
- Mori, S., Chang, K., Chong, C. & Dunn, K. (1986), ‘Prediction of track purity and track accuracy in dense target environments’, *IEEE Transactions on Automatic Control* **40**, 953–959.
- Obermeyer, F. H. & Poore, A. (2004), ‘A Bayesian network tracking database’, *Proceedings of the SPIE, ser. Conference on Signal and Data Processing of Small Targets* **5428**, 400–418.
- Pasquero, C., Provenazale, A. & Weiss, J. B. (2002), ‘Vortex statistics from Eulerian and Lagrangian time series’, *Physical Review Letters* **89**, 284501.
- Pece, A. (2002), ‘Generative-model-based tracking by cluster analysis of image differences’, *Robotics and Autonomous Systems* **39**, 181–194.
- Reid, D. B. (1979), ‘An algorithm for tracking multiple targets’, *IEEE Transactions on Automatic Control* **24**, 843–854.
- Roh, H., Kang, S. & Lee, S. (2000), ‘Multiple people tracking using an appearance model based on temporal color’, *Proceedings. 15th International Conference on Pattern Recognition* **4**, 643–646.

- Rosenfeld, A. & Kak, A. (1982), *Digital Picture Processing*, 2nd edn, Academic Press.
- Salari, V. & Sethi, I. K. (1990), ‘Feature point correspondence in the presence of occlusion’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, 87–91.
- Salmond, D. J. & Birch, H. (2001), ‘A particle filter for track-before-detect’, *Proceedings of the American Control Conference* **5**, 3755–3760.
- Salmond, D. J. & Parr, M. C. (2003), ‘Track maintenance using measurements of target extent’, *IEE Proceedings on Radar, Sonar, and Navigation* **150**, 389–395.
- Särkkä, S., Vehtari, A. & Lampinen, J. (2007), ‘Rao-Blackwellized particle filter for multiple target tracking’, *Information Fusion*. to appear.
- Schimek, M. (2000), *Smoothing and Regression: Approaches, Computation, and Application*, New York: John Wiley & Sons, Inc.
- Sethi, I. K. & Jain, R. (1987), ‘Finding trajectories of feature points in a monocular image sequence’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**, 56–72.
- Stone, L. D., Barlow, C. A. & Corwin, T. L. (1999), *Bayesian Multiple Target Tracking*, Boston: Artech House.
- Storlie, C. B., Davis, C., Hoar, T., Lee, T. C. M., Nychka, D., Weiss, J. & Whitcher, B. (2004), ‘Identifying and tracking turbulence structures’, *Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers* **2**, 1700–1704.
- Storlie, C. B., Hannig, J. & Lee, T. C. M. (2007), ‘The asymptotic properties of a multiple target tracking model’. Submitted for publication.
- Tonissen, S. M. & Bar-Shalom, Y. (1998), ‘Maximum likelihood track-before-detect with fluctuating target amplitude’, *IEEE Transactions on Aerospace and Electronic Systems* **34**, 796–809.
- Trunk, G. V. & Wilson, J. D. (1981), ‘Track initiation of occasionally unresolved radar targets’, *IEEE Transactions on Aerospace and Electronic Systems* **17**, 122–130.
- Tuttle, J. & Gall, R. (1999), ‘A single-radar technique for estimating the winds in tropical cyclones’, *Bulletin of the American Meteorological Society* **80**, 653–668.
- Vermaak, J., Godsill, S. & Perez, P. (2005), ‘Monte Carlo filtering for multitarget tracking and data association’, *IEEE Transactions on Aerospace and Electronic Systems* **41**, 309–332.
- Vo, B., Singh, S. & Doucet, A. (2005), ‘Sequential Monte Carlo methods for Bayesian multitarget filtering with random finite sets’, *IEEE Transactions on Aerospace and Electronic Systems* **41**, 1224–1245.
- Wang, H. & Lee, T. C. M. (2006), ‘Automatic parameter selection for a k-segments algorithm for computing principal curves’, *Pattern Recognition Letters* **27**, 1142–1150.
- Wang, Y. & Zhu, S. (2004), ‘Analysis and synthesis of textured motion: Particles and waves’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1348–1363.
- Weiss, J. B. & McWilliams, J. C. (1993), ‘Temporal scaling behavior of decaying two-dimensional turbulence’, *Physics of Fluids A: Fluid Dynamics* **5**, 608–621.

Wolfson, M., Forman, B., Hallowell, R. & Moore, M. (1999), 'The growth and decay storm tracker', *8th Conference on Aviation, American Meteorological Society* pp. 58–62.

SUPPLEMENTARY DOCUMENT TO ACCOMPANY THE ARTICLE "TRACKING OF MULTIPLE MERGING AND SPLITTING TARGETS WITH APPLICATION TO CONVECTIVE SYSTEMS"

Curtis B. Storlie^a, Thomas C. M. Lee^{b,c}, Jan Hannig^c, and Douglas Nychka^d

^aDepartment of Mathematics and Statistics, University of New Mexico

^bDepartment of Statistics, The Chinese University of Hong Kong

^cDepartment of Statistics, Colorado State University

^dGeophysical Statistics Project, National Center for Atmospheric Research

This document provides many of the technical details of the model proposed in the main article in addition to some more in-depth discussion.

S.1 Calculation of Model Likelihood

In this Section we present the likelihood of the model described in Section 3. We will use the notation $[X]$ to denote the probability density function of the random variable X , $[X](x)$ to denote $[X]$ evaluated at x and $[X | Y]$ to denote the conditional density of X given Y . We wish to write out the density, or likelihood, for the following collection of random variables that correspond to targets,

$$(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}).$$

The bold \mathcal{W} , \mathcal{X} , and \mathcal{Y} denote the collection of those variables for all targets at all times. These variables will be more formally defined in the following sections. For ease of presentation, we first restrict the focus to location information. We will incorporate the attribute contribution to the likelihood later.

In addition, we wish to write out a density for the following collection of random variables that correspond to false alarms

$$(\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f), \tag{S.1}$$

where $\mathbf{N}_f = (N_f(t_1), \dots, N_f(t_n))$ and the bold \mathcal{X}_f and \mathcal{Y}_f denote the collection of the locations for all false alarms at all times. The overall model likelihood function is then given by

$$[(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}), (\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f)] = [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}][\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f],$$

as the false alarms are assumed to be completely independent of the targets. In the following sections we will write out the target density, $[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}]$, and the false alarm density, $[\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f]$.

S.1.1 Target Density

Since \mathcal{X} and \mathcal{Y} are independent given $(\mathcal{U}, \mathcal{V}, \mathcal{W})$, we can write the target density as

$$[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}] = [\mathcal{U}, \mathcal{V}] \cdot [\mathcal{W} \mid \mathcal{U}, \mathcal{V}] \cdot [\mathcal{X} \mid \mathcal{U}, \mathcal{V}, \mathcal{W}] \cdot [\mathcal{Y} \mid \mathcal{U}, \mathcal{V}, \mathcal{W}]. \quad (\text{S.2})$$

We will call the conditional densities in (S.2), in order from left to right, the target event density, observability density, and target location densities respectively. We will describe each of these in the following sections.

S.1.1.1 Target Event Density

Since the Event Model has independent increments, the Event density can be written as

$$[\mathcal{U}, \mathcal{V}] = [N_0] \prod_{j=1}^n [U_{b,j}, U_{d,j}, U_{s,j}, U_{m,j} \mid N(t_j)] [V_{b,j}, V_{d,j}, V_{s,j}, V_{m,j} \mid N(t_j), U_{b,j}, U_{d,j}, U_{s,j}, U_{m,j}], \quad (\text{S.3})$$

where recall that $N(t)$ is the number of targets that exist at time t . Also, N_0 is the initial number of targets and is assumed to be Poisson distributed with parameter λ_0 . Therefore

$$[N_0](k) = \frac{\lambda_0^k e^{-\lambda_0}}{k!}.$$

To write out the exact density for $(U_{b,j}, U_{d,j}, U_{s,j}, U_{m,j} \mid N(t_j))$ is difficult since they are dependent on each other. The rate of death, $\lambda_d N(t)$, for example changes when there is a birth, death, split or

merger. Suppose $U_j = U_{b,j} + U_{d,j} + U_{s,j} + U_{m,j}$. The exact distribution of $(U_{b,i}, U_{d,i}, U_{s,i}, U_{m,i})$ would require us to sum over all the permutations of the order that the U_j events could happen in the interval $[t_j, t_{j+1})$. For each of these permutations, we would have to calculate the probability that the sum of U_j independent exponential random variables with respective rates (which are generally different) would be less than $\Delta t_j = t_{j+1} - t_j$. Instead, we will approximate this probability by assuming that the rate of the occurrence of events stays constant during the interval $[t_j, t_{j+1})$. If we let $\bar{N}_j = (N(t_j) + N(t_{j+1}))/2$, which is the average number of targets alive during the interval, then we can assume that the rate of each of the events during the interval is $\bar{\lambda}_{b,j} = \lambda_b$, $\bar{\lambda}_{d,j} = \lambda_d \bar{N}_j$, $\bar{\lambda}_{s,j} = \lambda_s \bar{N}_j$ and $\bar{\lambda}_{m,j} = \lambda_m (\bar{N}_j - 1)$ for birth, death, splitting, and merging events respectively. With this assumption, the variables $(U_{b,i}, U_{d,i}, U_{s,i}, U_{m,i})$ are independent and $P(U_{d,j} = u)$ for example is the probability that the sum of u iid exponential random variables with rate $\bar{\lambda}_{d,j}$ are less than Δt_j . This is the same as the Poisson density with parameter $\bar{\lambda}_{d,j} \Delta t_j$ evaluated at u . Hence,

$$\begin{aligned}
[U_{b,j} \mid N(t_j)](u) &\approx (\lambda_b \Delta t_j)^u e^{-\lambda_b \Delta t_j} / u! \\
[U_{d,j} \mid N(t_j)](u) &\approx (\bar{\lambda}_{d,j} \Delta t_j)^u e^{-\bar{\lambda}_{d,j} \Delta t_j} / u! \\
[U_{s,j} \mid N(t_j)](u) &\approx (\bar{\lambda}_{s,j} \Delta t_j)^u e^{-\bar{\lambda}_{s,j} \Delta t_j} / u! \\
[U_{m,j} \mid N(t_j)](u) &\approx (\bar{\lambda}_{m,j} \Delta t_j)^u e^{-\bar{\lambda}_{m,j} \Delta t_j} / u!
\end{aligned} \tag{S.4}$$

Under the same assumption that $N(t) = \bar{N}_j$ is constant during the interval $[t_j, t_{j+1})$, we have

$$\begin{aligned}
[V_{b,j} \mid N(t_j), U_{b,j}](v) &\approx 1 \\
[V_{d,j} \mid N(t_j), U_{d,j}](v) &\approx (1/\bar{N}_j)^{U_{d,j}} \\
[V_{s,j} \mid N(t_j), U_{s,j}](v) &\approx (1/\bar{N}_j)^{U_{s,j}} \\
[V_{m,j} \mid N(t_j), U_{m,j}](v) &\approx \left(1/\binom{\bar{N}_j}{2}\right)^{U_{m,j}}
\end{aligned}$$

and we can write (S.3) as

$$[\mathcal{U}, \mathcal{V}] \approx [N_0] \prod_{j=1}^n [U_{b,j}] [V_{b,j} \mid U_{b,j}] \cdot [U_{d,j}] [V_{d,j} \mid U_{d,j}] \cdot [U_{s,j}] [V_{s,j} \mid U_{s,j}] \cdot [U_{m,j}] [V_{m,j} \mid U_{m,j}]. \quad (\text{S.5})$$

S.1.1.2 Observability Density

Recall that $W_i(t)$ represents the observability (0 or 1) of the i^{th} target at time t , $i = 1, \dots, M$ where M is the number of targets that existed before time t_n . Let $\mathcal{W} = \{W_i(t_j) : i = 1, \dots, M, j = 1, \dots, n\}$. The time of initiation of the i^{th} target is denoted by ξ_i . Also let the time of termination of the i^{th} target be given by ζ_i . For convenience if the i^{th} target is still alive at time t_n , we will let $\zeta_i = t_n$.

The events variables \mathcal{U} and \mathcal{V} do not specify the exact values of ξ_i and ζ_i . They do however specify which interval between observations they are in. This completely specifies \mathcal{W} since its dependence on \mathcal{U} and \mathcal{V} is only on whether or not a target exists at the observed time points. In the sequel, if it is known that ξ_i is in the interval (t_j, t_{j+1}) , we will set $\xi_i = t_j + \Delta t_j / 2$.

The white noise model for \mathcal{W} of Section 3.2 assumes probability P_d of observing the i^{th} target if it exists at a given time, independent of other times. If the target does not exist at time t then $W_i(t) = 0$. Under this model, the conditional density of \mathcal{W} given the event variables in (S.2) can be written out using indicator functions to separate the cases when the i^{th} target exists and when it does not. This density is then given by

$$[\mathcal{W} \mid \mathcal{U}, \mathcal{V}](w) = \prod_{i=1}^M \prod_{j=1}^n \left\{ I_{[t_1, \xi_i) \cup (\zeta_i, t_n]}(t_j) (1 - w_{ij}) + I_{[\xi_i, \zeta_i]}(t_j) ((1 - w_{ij})(1 - P_d) + w_{ij} P_d) \right\},$$

where w_{ij} is representing an observed value of $W_i(t_j)$.

S.1.1.3 Target Location Density

Since $X_i(t)$ is normally distributed for all t , the observed location of all targets at all time points has a multivariate normal distribution. Let the times at which the i^{th} target is observable be denoted by $\mathbf{t}_i = (t_{i,1}, \dots, t_{i,n_i})$. Also let $\mathbf{X}_i = (X_i(t_{i,1}), \dots, X_i(t_{i,n_i}))'$ and lastly let $\mathcal{X} = (\mathbf{X}'_1, \dots, \mathbf{X}'_M)'$ be the collection of all observed locations of all targets during the time sequence t_1, \dots, t_n . Then

$\mathcal{X} \sim \mathcal{N}(\mu_X, \Sigma_X)$, where we will define μ_X , and Σ_X below.

Recall from Section 3.3 that this mean and covariance will depend on the time of initiation, ξ , of the targets. We will adopt the convention of the previous section here and set $\xi_i = t_j + \Delta t_j/2$ if ξ_i is known to be in the interval (t_j, t_{j+1}) . Since μ_X and Σ_X depend on the exact values of ξ , this will be an approximation to the true density. In order to calculate the exact density, we would need to integrate out on the joint distribution of \mathcal{X} and ξ , given that the ξ_i 's are in their respective intervals. Most likely this can only be achieved via numerical approximations.

Also recall from Section 3.3 that we need to condition \mathcal{X} on the random variables (D_1, \dots, D_{N_m}) and evaluate this density when they are zeros. Let $\mathcal{D} = (D_1, \dots, D_{N_m})'$, and we write

$$\mathcal{D} \sim \mathcal{N}(\mu_D, \Sigma_D).$$

For the collection of both \mathcal{X} and \mathcal{D} we have

$$\begin{pmatrix} \mathcal{X} \\ \mathcal{D} \end{pmatrix} \sim \mathcal{N}(\mu, \Sigma) \quad (\text{S.6})$$

where

$$\mu = \begin{pmatrix} \mu_X \\ \mu_D \end{pmatrix} \quad (\text{S.7})$$

and

$$\Sigma = \left(\begin{array}{c|c} \Sigma_X & \Sigma_{X,D} \\ \hline \Sigma'_{X,D} & \Sigma_D \end{array} \right).$$

The mean vectors and covariance matrices will be described in the following. Let $\mu_i(t) = E\{X_i(t)\}$ and $\mu_{D_i} = E(D_i)$. These functions are given for the IBM model in Section S.2. Then let $\boldsymbol{\mu}_i = (\mu_i(t_{i,1}), \dots, \mu_i(t_{i,n_i}))$ and we can now write the mean vectors in (S.7) as $\mu_X = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m)$ and $\mu_D = (\mu_{D_1}, \dots, \mu_{D_{N_m}})$.

Define the matrices $\Sigma_{i,j}$ to be the covariances between all of the observations of target path i with all of the observations of path j . Specifically the $(k, l)^{th}$ element of this matrix can be written as

$$\Sigma_{i,j}(k, l) = \text{Cov}(X_i(t_{i,k}), X_j(t_{j,l})), \quad k = 1, \dots, n_i; \quad l = 1, \dots, n_j. \quad (\text{S.8})$$

Also define the matrices $\Sigma_{i,D}$ and Σ_D by their $(k,l)^{th}$ element as follows

$$\Sigma_{i,D}(k,l) = \text{Cov}(X_i(t_{i,k}), D_l) \quad k = 1, \dots, n_i; \quad l = 1, \dots, N_m \quad (\text{S.9})$$

$$\Sigma_D(k,l) = \text{Cov}(D_k, D_l) \quad k = 1, \dots, N_m; \quad l = 1, \dots, N_m. \quad (\text{S.10})$$

The covariance functions in (S.8), (S.9), and (S.10) for the IBM model are given in Section S.2.

Now we can write the covariance matrix for \mathcal{X} as

$$\Sigma_X = \begin{pmatrix} \Sigma_{1,1} & \Sigma_{1,2} & \cdots & \Sigma_{1,m} \\ \Sigma_{2,1} & \Sigma_{2,2} & \cdots & \Sigma_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m,1} & \Sigma_{m,2} & \cdots & \Sigma_{m,m} \end{pmatrix}$$

and that for $(\mathcal{X}, \mathcal{D})$ as

$$\Sigma_{X,D} = \begin{pmatrix} \Sigma_{1,D} \\ \vdots \\ \Sigma_{m,D} \end{pmatrix}$$

This completes the description of the distribution of $(\mathcal{X}, \mathcal{D})$ given in (S.6).

We can then compute the conditional distribution of \mathcal{X} given $\mathcal{D} = 0$, which we will just call the distribution of \mathcal{X} from this point onward. From standard multivariate normal theory we have

$$\mathcal{X} \mid \mathcal{D} = 0 \sim \mathcal{N}(\mu^*, \Sigma^*)$$

where

$$\mu^* = \mu_X - \Sigma_{X,D} \Sigma_D^{-1} \mu_D \quad \text{and} \quad \Sigma^* = \Sigma_X - \Sigma_{X,D} \Sigma_D^{-1} \Sigma'_{X,D}.$$

The density of X is then just the multivariate normal density with parameters μ^* and Σ^* . This will require computing the inverse of Σ^* , which can be done quite efficiently since Σ^* is a relatively sparse matrix. Unless path i is a relative of path j , in the sense that one is a by-product of splitting or merging of the other, they will have 0 covariance. Unfortunately, because of the conditioning on \mathcal{D} , this model cannot be posed in state space form. Hence, the corresponding filters cannot be used to update the conditional distribution of a new observation given the previous observations.

S.1.2 False Alarm Density

In a manner similar to target density we can write the density of the false alarm variables from (S.1) as

$$[\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f] = [\mathbf{N}_f] \cdot [\mathcal{X}_f, \mathcal{Y}_f \mid \mathbf{N}_f] \quad (\text{S.11})$$

where \mathbf{N}_f , \mathcal{X}_f , and \mathcal{Y}_f will be precisely defined below. It was assumed in Section 3.5 that false alarms occur at each time frame as *iid* realizations from a Poisson Process with intensity function $\rho(x, y)$. Hence $\mathbf{N}_f = (N_f(t_1), \dots, N_f(t_n))$ are *iid* Poisson distributed random variables with rate $\lambda_f = \int \rho(x, y) dx dy$. The corresponding density of \mathbf{N}_f is then

$$[\mathbf{N}_f](\mathbf{k}) = \prod_{j=1}^n \frac{\lambda_f^{k_j} e^{-\lambda_f}}{k_j!}.$$

Now let the x component of the i^{th} false alarm at time t be denoted as $X_{f,i}(t)$ for $i = 1, \dots, N_f(t)$. Also let $\mathcal{X}_f = \{X_{f,i}(t_j) : i = 1, \dots, N_f(t_j), j = 1, \dots, n\}$ be the collection of x locations of all false alarms at all times. Similar notation will be used for \mathcal{Y}_f . Due to the Poisson process assumption, the density function for a particular $(X_{f,i}(t), Y_{f,i}(t))$ is $f(x, y) = \rho(x, y)/\lambda_f$ and hence the density for $(\mathcal{X}_f, \mathcal{Y}_f)$ is

$$[\mathcal{X}_f, \mathcal{Y}_f \mid \mathbf{N}_f](x) = \prod_{j=1}^n \prod_{i=1}^{N_f(t_j)} \rho(x_{ij}, y_{ij})/\lambda_f,$$

where x_{ij} is a dummy variable for the value of $X_{f,i}(t_j)$ and similarly for y_{ij} .

S.1.3 Attributes

The attribute variables are assumed *iid* over time given the Observability variable \mathcal{W} , thus the densities are quite straightforward to calculate. With the presence of attributes, we now have the following collection of random variables for targets

$$(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}),$$

where \mathcal{A} denotes the collection of all attribute variables. Below we will assume $\mathcal{A} = (\mathcal{R}_{(1)}, \mathcal{R}_{(2)}, \mathcal{Q}_{(2)}, \mathcal{I})$, which are the smallest radius, largest radius, angle of orientation and intensity for targets respectively. These variables will be formally defined later.

We also have the following collection of random variables that correspond to false alarms:

$$(\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f),$$

where $\mathcal{A}_f = (\mathcal{R}_{(1),f}, \mathcal{R}_{(2),f}, \mathcal{Q}_{(2),f}, \mathcal{I}_f)$, which are the same variables as above but for false alarms.

The target likelihood function is then given by

$$[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}] = [\mathcal{U}, \mathcal{V}] \cdot [\mathcal{W} \mid \mathcal{U}, \mathcal{V}] \cdot [\mathcal{X} \mid \mathcal{U}, \mathcal{V}, \mathcal{W}] \cdot [\mathcal{Y} \mid \mathcal{U}, \mathcal{V}, \mathcal{W}] \cdot [\mathcal{A} \mid \mathcal{W}].$$

So we can just multiply $[\mathcal{A} \mid \mathcal{W}]$ to the target density without attributes given in (S.2). Technically \mathcal{A} should also be conditioned on \mathcal{U} , \mathcal{V} , \mathcal{X} , and \mathcal{Y} as well, but given the way that we modeled attributes in the previous section, the density of \mathcal{A} would still depend only on \mathcal{W} , and hence we dropped the other variables in the notation. Similarly, the false alarm likelihood is given by

$$[\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f] = [\mathbf{N}_f] \cdot [\mathcal{X}_f \mid \mathbf{N}_f] \cdot [\mathcal{Y} \mid \mathbf{N}_f] \cdot [\mathcal{A} \mid \mathbf{N}_f]$$

so we can just multiply $[\mathcal{A} \mid \mathbf{N}_f]$ to the false alarm density without attributes in (S.11). Therefore the overall density is

$$[(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}), (\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f)] = [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}] \cdot [\mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f]. \quad (\text{S.12})$$

We can of course incorporate any of these attribute variables separately or add other attributes in a similar manner. For the collection above though, we have

$$[\mathcal{A} \mid \mathcal{W}] = [\mathcal{R}_{(1)}, \mathcal{R}_{(2)} \mid \mathcal{W}][\mathcal{Q}_{(2)} \mid \mathcal{W}][\mathcal{I} \mid \mathcal{W}]$$

and

$$[\mathcal{A} \mid \mathbf{N}_f] = [\mathcal{R}_{(1),f}, \mathcal{R}_{(2),f} \mid \mathbf{N}_f][\mathcal{Q}_{(2),f} \mid \mathbf{N}_f][\mathcal{I}_f \mid \mathbf{N}_f].$$

We will describe these densities in the following sections.

S.1.3.1 Radius Density

Let $R_{1,i}(t)$ and $R_{2,i}(t)$ respectively be the length of minor and major axes of the best fitting ellipse to target i at time t . We only observe the min and max of these from the data which are $R_{(1),i}(t)$ and $R_{(2),i}(t)$ respectively. Also let

$$\mathcal{R}_{(1)} = \{(R_{(1)}(t_j) : 1 \leq i \leq M, 1 \leq j \leq n)\}$$

and similarly for $\mathcal{R}_{(2)}$, where recall M is the total number of targets that existed before time t_n .

Recall that $R_{1,i}(t)$ and $R_{2,i}(t)$ are assumed to be distributed as independent log-normals for all t . The density for $(R_{(1),i}(t), R_{(2),i}(t))$ does not depend on time so we will write it as $[R_{(1),i}, R_{(2),i}]$. This density is similar to that for order statistics and is given by

$$[R_{(1),i}, R_{(2),i}](r, s) = \{[R_{1,i}](r)[R_{2,i}](s) + [R_{1,i}](s)[R_{2,i}](r)\} I_{\{r \leq s\}} \quad (\text{S.13})$$

where $[R_{1,i}]$ and $[R_{2,i}]$ are log-normal densities with parameters $(\mu_{R_{1,i}}, \sigma_{R_{1,i}}^2)$ and $(\mu_{R_{2,i}}, \sigma_{R_{2,i}}^2)$ respectively as described in Section 3.4.

Since the radii of path i at time t_j are independent of the radii at other times or of other targets, the density for $(\mathcal{R}_{(1)}, \mathcal{R}_{(2)})$ is

$$[\mathcal{R}_{(1)}, \mathcal{R}_{(2)} \mid W](r, s) = \prod_{i=1}^M \prod_{\{j: W_{i,j}=1\}} [R_{(1),i}, R_{(2),i}](r_{i,j}, s_{i,j}),$$

where $r_{i,j}$ and $s_{i,j}$ are the arguments for the values of $R_{(1),i}(t_j)$ and $R_{(2),i}(t_j)$ respectively.

For false alarms, we will use similar notation. Let $(R_{1,f,i}(t)$ and $R_{2,f,i}(t))$ be the length of minor and major axes of the best fitting ellipse to the i^{th} false alarm at time t . We observe the min and

max of these which are $R_{(1),f,i}(t)$ and $R_{(2),f,i}(t)$ respectively. Also let

$$\mathcal{R}_{(1),f} = \{(R_{(1),i}(t_j) : 1 \leq j \leq n, \ 1 \leq i \leq N_f(t_j)\},$$

and similarly for $\mathcal{R}_{(2),f}$.

The density for false alarms is very similar to that above, but all false alarms at all times are assumed to have the same distribution so

$$[R_{(1),f,i}(t), R_{(2),f,i}(t)] = [R_{(1),f,i'}(t), R_{(2),f,i'}(t)] = [R_{(1),f}, R_{(2),f}]$$

where

$$[R_{(1),f}, R_{(2),f}](r, s) = \{[R_{1,f}](r)[R_{2,f}](s) + [R_{1,f}](s)[R_{2,f}](r)\} I_{\{r \leq s\}}$$

and $[R_{1,i}]$ and $[R_{2,i}]$ are respectively log-normal densities with parameters $(\mu_{R_{1,f}}, \sigma_{R_{1,f}}^2)$ and $(\mu_{R_{2,f}}, \sigma_{R_{2,f}}^2)$.

So the density of $(\mathcal{R}_{(1),f}, \mathcal{R}_{(2),f})$ is

$$[\mathcal{R}_{(1),f}, \mathcal{R}_{(2),f} \mid \mathbf{N}_f](r, s) = \prod_{j=1}^n \prod_{i=1}^{N_f(t_j)} [R_{(1),f}, R_{(2),f}](r_{i,j}, s_{i,j})$$

where $r_{i,j}$ and $s_{i,j}$ are the arguments for the values of $R_{(1),f,i}(t_j)$ and $R_{(2),f,i}(t_j)$ respectively.

S.1.3.2 Angle of Orientation Density

For target orientation or angle, we will use the following notation. Let $Q_{2,i}(t)$ be the angle of orientation of the axis corresponding to R_2 of the best fitting ellipse to target i at time t . We actually observe $Q_{(2),i}(t)$ which is the angle that corresponds to $R_{(2),i}(t)$. Also let

$$\mathcal{Q}_{(2)} = \{Q_{(2)}(t_j) : 1 \leq i \leq M, \ 1 \leq j \leq n\}.$$

Consider for now a given target's orientation at a fixed time $Q_{(2),i}(t)$. We will drop the subscript i and argument t for now and write this as $Q_{(2)}$ to make notation less cumbersome. When $R_{(2)} = R_2$, $Q_{(2)} = Q_2$. However, when $R_{(2)} = R_1$, $Q_{(2)} = \lfloor Q_2 + \pi/2 \rfloor_p i$ where $\lfloor x \rfloor_y$ is $x \bmod y$.

Hence, the distribution of $Q_{(2)}$ given $(R_{(1)}, R_{(2)})$ is a mixture distribution that takes the value of Q_2 with probability γ and $\lfloor Q_2 + \pi/2 \rfloor_\pi$ with probability $1 - \gamma$, where

$$\begin{aligned}\gamma &= P(R_1 < R_2 \mid R_{(1)}, R_{(2)}) \\ &= \frac{[R_1](R_{(1)})[R_2](R_{(2)})}{[R_1](R_{(1)})[R_2](R_{(2)}) + [R_1](R_{(2)})[R_2](R_{(1)})}.\end{aligned}\tag{S.14}$$

Thus the conditional density of $Q_{(2),i}$ is

$$[Q_{(2),i} \mid R_{(1)}, R_{(2)}](q) = \gamma[Q_{2,i}](q) + (1 - \gamma)[Q_{2,i}](\lfloor q + \pi/2 \rfloor_\pi),\tag{S.15}$$

where $[Q_{2,i}]$ is the von Mises density on $[0, \pi)$ given by

$$[Q_{2,i}](q) = \frac{e^{\beta_i \cos(q - \alpha_i)}}{\pi \Psi_0(\beta_i)} I_{[0, \pi)}(q).$$

Here $\Psi_0(x)$ is a modified Bessel function of the first kind of order 0. As with the radii, $Q_{(2),i}(t)$ is independent over time and of other targets so the conditional density of $\mathcal{Q}_{(2)}$ is

$$[\mathcal{Q}_{(2)} \mid \mathcal{W}, \mathcal{R}_{(1)}, \mathcal{R}_{(2)}](q) = \prod_{i=1}^M \prod_{\{j: W_{i,j}=1\}} [Q_{(2)} \mid R_{(1),i}(t_j), R_{(2),i}(t_j)](q_{i,j})\tag{S.16}$$

where $q_{i,j}$ are the arguments for the values of $Q_{(2),i}(t_j)$.

Again the situation for false alarms is very similar. We will let $Q_{(2),f,i}(t)$ be the angle of orientation corresponding to $R_{(2),f,i}(t)$ and

$$\mathcal{Q}_{(2),f} = \{Q_{(2),i}(t_j) : 1 \leq j \leq n, \ 1 \leq i \leq N_f(t_j)\}.$$

Let $[Q_{2,f}]$ be the same density as in (S.15) only with parameters α_f and β_f in place of α_i and β_i . False alarms are *iid* so

$$[\mathcal{Q}_{(2),f}(t) \mid \mathcal{W}, \mathcal{R}_{(1)}, \mathcal{R}_{(2)}](q) \prod_{j=1}^n \prod_{i=1}^{N_f(t_j)} [Q_{(2),f} \mid R_{(1),f,i}(t_j), R_{(2),f,i}(t_j)](q_{i,j}).$$

S.1.3.3 Intensity Density

Let $I_i(t)$ be the intensity of target i at time t . Also let

$$\mathcal{I} = \{I_i(t_j) : 1 \leq i \leq M, 1 \leq j \leq n\}.$$

For any target the density of $I_i(t)$ does not depend on time so we will write it as $[I_i]$. Recall from Section 3.4 that $[I_i]$ is assumed to be a log-normal density with parameters $(\mu_{I_i}, \sigma_{I_i}^2)$. The density of \mathcal{I} is then

$$[\mathcal{I} \mid \mathcal{W}](\iota) \prod_{i=1}^M \prod_{\{j: W_{i,j}=1\}} [I_i](\iota_{i,j})$$

where as usual $\iota_{i,j}$ are the arguments for the values of $I_i(t_j)$.

For false alarm intensity, we again assume the same density $[I_f]$ for all false alarms which is log-normal with parameters $(\mu_{I_f}, \sigma_{I_f}^2)$. The density of \mathcal{I}_f is then

$$[\mathcal{I} \mid \mathcal{W}](\iota) \prod_{j=1}^n \prod_{i=1}^{N_f(t_j)} [I_f](\iota_{i,j}).$$

S.2 Mean and Covariance Calculations

Here we calculate the mean functions $E\{X_i(t)\}$, $E(D_i)$ and the covariance functions $\text{Cov}(X_i(s), X_j(t))$, $\text{Cov}(X_i(s), D_j)$, and $\text{Cov}(D_i, D_j)$. Recall in Section S.1.1.3 that these are calculated before conditioning on any merging events.

Let

$$\mathcal{B} = \{i : \text{target } i \text{ is an initial target or a birth}\}$$

$$\mathcal{S} = \{i : \text{target } i \text{ is the result of a splitting event}\}$$

$$\mathcal{M} = \{i : \text{target } i \text{ is the result of a merging event}\}$$

Also let $n(\mathcal{B})$, $n(\mathcal{S})$ and $n(\mathcal{M})$ be the number elements in these sets respectively. The location equations for a target resulting from birth, splitting and merging events from (3), (7) and (5) are

given here again for convenient reference as

$$X_i(t) = \begin{cases} X_i(\xi_i) + X'_i(\xi_i)(t - \xi_i) + \sigma_i G_i(t - \xi_i) & \text{for } i \in \mathcal{B} \\ X_{p_{i,1}}(\xi_i) + \psi_{s,i} + \left[X'_{p_{i,1}}(\xi_i) + \psi'_{s,i} \right] (t - \xi_i) + \sigma_i G_i(t - \xi_i) & \text{for } i \in \mathcal{S} \\ \frac{1}{2} (X_{p_{i,1}}(\xi_i) + X_{p_{i,2}}(\xi_i)) + \psi_{m,i} + \left[\frac{1}{2} (X'_{p_{i,1}}(\xi_i) + X'_{p_{i,2}}(\xi_i)) + \psi'_{m,i} \right] (t - \xi_i) + \sigma_i G_i(t - \xi_i) & \text{for } i \in \mathcal{M} \end{cases}$$

where we are assuming that $G_i(t)$ is an IBM. Also recall that we actually observe

$X_i^*(t_j) = X_i(t_j) + \varepsilon_j$ for each time point t_j . We give the target velocities for the three cases as well,

$$X'_i(t) = \begin{cases} X'_i(\xi_i) + \sigma_i B_i(t - \xi_i) & \text{for } i \in \mathcal{B} \\ X'_{p_{i,1}}(\xi_i) + \psi'_{s,i} + \sigma_i B_i(t - \xi_i) & \text{for } i \in \mathcal{S} \\ \frac{1}{2} (X'_{p_{i,1}}(\xi_i) + X'_{p_{i,2}}(\xi_i)) + \psi'_{m,i} + \sigma_i B_i(t - \xi_i) & \text{for } i \in \mathcal{M}. \end{cases}$$

Lastly we recall the expression for the variable $D_i = X_{d_{i,1}}(\xi_{d_{i,3}}) - X_{d_{i,2}}(\xi_{d_{i,3}}) + \psi_{d,i}$, $i = 1, \dots, n(\mathcal{M})$.

We will use the following notation to denote the means and covariance of path locations and velocities

$$\mu_i(t) = E\{X_i(t)\} \tag{S.17}$$

$$\mu'_i(t) = E\{X'_i(t)\}$$

$$\gamma_{i,j}^*(s, t) = \text{Cov}(X_i^*(s), X_i^*(t)) \tag{S.18}$$

$$\gamma_{i,j}(s, t) = \text{Cov}(X_i(s), X_j(t))$$

$$\gamma'_{i,j}(s, t) = \text{Cov}(X_i(s), X'_j(t))$$

$$\gamma''_{i,j}(s, t) = \text{Cov}(X'_i(s), X'_j(t))$$

$$\gamma_i(s, t) = \text{Cov}(X_i(s), X_i(t))$$

$$\gamma'_i(s, t) = \text{Cov}(X_i(s), X'_i(t))$$

$$\gamma''_i(s, t) = \text{Cov}(X'_i(s), X'_i(t))$$

Note that for the purposes of likelihood calculation, we are only interested in the functions

given in (S.17) and (S.18) above. However, the expressions for these two functions will depend on the others, so in the following, we will need to derive expressions for all of these functions.

S.2.1 Mean Functions

We can express the mean functions for location for the three cases of birth, splitting and merging events as a recursive formula,

$$\mu_i(t) = \begin{cases} \mu_{X_0} + (t - \xi_i)\mu_{X'_0} & \text{if } i \in \mathcal{B} \\ \mu_{p_{i,1}}(\xi_i) + (t - \xi_i)\mu'_{p_{i,1}}(\xi_i) & \text{if } i \in \mathcal{S} \\ \frac{1}{2}\mu_{p_{i,1}}(\xi_i) + \frac{1}{2}\mu_{p_{i,2}}(\xi_i) + \frac{t-\xi_i}{2} \left(\mu'_{p_{i,1}}(\xi_i) + \mu'_{p_{i,2}}(\xi_i) \right) & \text{if } i \in \mathcal{M}. \end{cases}$$

Eventually this recursion will lead back to a parent target which is an initial target or a birth, at which point the recursion will terminate. We can also express the mean velocities for the three cases in a similar manner,

$$\mu_i(t) = \begin{cases} \mu_{X'_0} & \text{if } i \in \mathcal{B} \\ \mu'_{p_{i,1}}(\xi_i) & \text{if } i \in \mathcal{S} \\ \frac{1}{2} \left(\mu'_{p_{i,1}}(\xi_i) + \mu'_{p_{i,2}}(\xi_i) \right) & \text{if } i \in \mathcal{M}. \end{cases}$$

Of course the mean of D_i can be written as

$$E(D_i) = \mu_{d_{i,1}}(\xi_{d_{i,3}}) - \mu_{d_{i,2}}(\xi_{d_{i,3}}).$$

S.2.2 Covariance Functions

Now we will consider the calculation of the covariance functions. First note that

$$\gamma_{i,j}^*(s, t) = \gamma_{i,j}(s, t) + \sigma_{X_e}^2 I_{\{i=j\}} I_{\{s=t\}}.$$

Also, for $\text{Cov}(X_i(s), D_j)$ and $\text{Cov}(D_i, D_j)$ we have

$$\begin{aligned}\text{Cov}(X_i(s), D_j) &= \gamma_{i,d_{j,1}}(s, \xi_{d_{j,3}}) - \gamma_{i,d_{j,2}}(s, \xi_{d_{j,3}}) \\ \text{Cov}(D_i, D_j) &= \gamma_{d_{i,1},d_{j,1}}(\xi_{d_{i,3}}, \xi_{d_{j,3}}) - \gamma_{d_{i,1},d_{j,2}}(\xi_{d_{i,3}}, \xi_{d_{j,3}}) - \gamma_{d_{i,2},d_{j,1}}(\xi_{d_{i,3}}, \xi_{d_{j,3}}) + \\ &\quad \gamma_{d_{i,2},d_{j,2}}(\xi_{d_{i,3}}, \xi_{d_{j,3}}),\end{aligned}$$

so we just need to derive an expression for $\gamma_{i,j}(s, t)$. This will require the following definition. Let two paths i and j be *connected* if one is a by-product of a splitting and/or merging of the other. Define the indicator $\delta_{i,j}$ to be

$$\delta_{i,j} = \begin{cases} 1 & \text{if path } i \text{ is connected to path } j \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that $\gamma_{i,j}(s, t) = 0$ whenever $\delta_{i,j} = 0$, since paths are independent unless they are connected.

Consider now calculating the covariance function $\gamma_{i,j}(s, t)$ when $\delta_{i,j} = 1$, $i < j$ and $j \in \mathcal{S}$:

$$\begin{aligned}\gamma_{i,j}(s, t) &= \text{Cov}\left(X_i(s), X_{p_{j,1}}(\xi_j) + \psi_{s,j} + \left[X'_{p_{j,1}}(\xi_j) + \psi'_{s,j}\right](t - \xi_j) + \sigma_j G_j(t - \xi_j)\right) \\ &= \gamma_{i,p_{j,1}}(s, \xi_j) + (t - \xi_j) \gamma'_{i,p_{j,1}}(s, \xi_j).\end{aligned}\tag{S.19}$$

Using this same idea, we can calculate the case for $\delta_{i,j} = 1$, $i < j$ and $j \in \mathcal{M}$ as well. If $i < j$ and $j \in \mathcal{B}$ then necessarily $\delta_{i,j} = 0$. This is true because if $i < j$ and $j \in \mathcal{B}$, then because of the way we have organized the indices, $\xi_i \leq \xi_j$. Hence if $j \in \mathcal{B}$ then there is now way that path j or any of its children could have split or merged to create path i since it existed already before path j . Furthermore, path j resulted from a birth so there is also no way that it could be created from path i or any of its children. Since we always decompose the larger index into the contribution from its parents, we will eventually converge to the covariance of a parent(s) that is a birth or initial target and the recursion will terminate.

Hence we have

$$\gamma_{i,j}(s,t) = \begin{cases} \gamma_i(s,t) & \text{if } i = j \\ \gamma_{i,p_{j,1}}(s, \xi_i) + (t - \xi_i) \gamma'_{i,p_{j,1}}(s, \xi_i) & \text{if } \delta_{i,j} = 1, i < j, j \in \mathcal{S} \\ \frac{1}{2} (\gamma_{i,p_{j,1}}(s, \xi_i) + \gamma_{i,p_{j,2}}(s, \xi_i)) + \frac{t - \xi_i}{2} (\gamma'_{i,p_{j,1}}(s, \xi_i) + \gamma'_{i,p_{j,2}}(s, \xi_i)) & \text{if } \delta_{i,j} = 1, i < j, j \in \mathcal{M} \\ \gamma_{j,i}(t, s) & \text{if } \delta_{i,j} = 1, i > j \\ 0 & \text{otherwise.} \end{cases} \quad (\text{S.20})$$

We can also calculate $\gamma'_{i,j}(s, t)$ in the same manner as in (S.19). Although, now we cannot use the symmetry of the function if $i > j$. So consider calculating $\gamma'_{i,j}(s, t)$ for the case when $\delta_{i,j} = 1$, $i > j$ and $i \in \mathcal{S}$. We still need to decompose the larger index into its parents, and we write this as

$$\begin{aligned} \gamma'_{i,j}(s, t) &= \text{Cov} \left(X_{p_{i,1}}(\xi_i) + \psi_{s,i} + [X'_{p_{i,1}}(\xi_i) + \psi'_{s,i}] (s - \xi_i) + \sigma_i G_i(s - \xi_i), X'_j(t) \right) \\ &= \gamma'_{p_{i,1},j}(\xi_i, t) + (s - \xi_i) \gamma''_{p_{i,1},j}(\xi_i, t). \end{aligned}$$

The other cases are similar and $\gamma'_{i,j}(s, t)$ can be written as

$$\gamma'_{i,j}(s, t) = \begin{cases} \gamma'_i(s, t) & \text{if } i = j \\ \gamma'_{i,p_{j,1}}(s, \xi_i) & \text{if } \delta_{i,j} = 1, i < j, j \in \mathcal{S} \\ \frac{1}{2} (\gamma'_{i,p_{j,1}}(s, \xi_i) + \gamma'_{i,p_{j,2}}(s, \xi_i)) & \text{if } \delta_{i,j} = 1, i < j, j \in \mathcal{M} \\ \gamma'_{p_{i,1},j}(\xi_i, t) + (s - \xi_i) \gamma''_{p_{i,1},j}(\xi_i, t) & \text{if } \delta_{i,j} = 1, i > j, j \in \mathcal{S} \\ \frac{1}{2} (\gamma'_{p_{i,1},j}(\xi_i, t) + \gamma'_{p_{i,2},j}(\xi_i, t)) + \frac{s - \xi_i}{2} (\gamma''_{p_{i,1},j}(\xi_i, t) + \gamma''_{p_{i,2},j}(\xi_i, t)) & \text{if } \delta_{i,j} = 1, i > j, j \in \mathcal{M} \\ 0 & \text{otherwise.} \end{cases}$$

We can calculate $\gamma''_{i,j}(s, t)$ in the same way as in (S.20), since we again have symmetry in the

function:

$$\gamma''_{i,j}(s,t) = \begin{cases} \gamma''_i(s,t) & \text{if } i = j \\ \gamma''_{i,p_{j,1}}(s, \xi_i) & \text{if } \delta_{i,j} = 1, i < j, j \in \mathcal{S} \\ \frac{1}{2} \left(\gamma''_{i,p_{j,1}}(s, \xi_i) + \gamma''_{i,p_{j,2}}(s, \xi_i) \right) & \text{if } \delta_{i,j} = 1, i < j, j \in \mathcal{M} \\ \gamma''_{j,i}(t, s) & \text{if } \delta_{i,j} = 1, i > j \\ 0 & \text{otherwise.} \end{cases}$$

Now for the function $\gamma_i(s, t)$. We can use the same technique in (S.19) but decompose both arguments to the covariance since they are the same path. For example, if target i is a birth or an initial target, then we have

$$\begin{aligned} \gamma_i(s, t) &= \text{Cov}\left(X_i(\xi_i) + X'_i(\xi_i)(s - \xi_i) + \sigma_i G_i(s - \xi_i), X_i(\xi_i) + X'_i(\xi_i)(t - \xi_i) + \sigma_i G_i(t - \xi_i)\right) \\ &= \sigma_{X_0}^2 + (s - \xi_i)(t - \xi_i)\sigma_{X'_0}^2 + \sigma_i^2 \text{Cov}(G_i(s - \xi), G_i(t - \xi)), \end{aligned} \quad (\text{S.21})$$

where for an IBM

$$\text{Cov}(G_i(s), G_i(t)) = \frac{(s \wedge t)^2(s \vee t)}{2} - \frac{(s \wedge t)^3}{6}.$$

If target i is a split then we have

$$\begin{aligned} \gamma_i(s, t) &= \text{Cov}\left(X_{p_{i,1}}(\xi_i) + \psi_{s,i} + \left[X'_{p_{i,1}}(\xi_i) + \psi'_{s,i}\right](s - \xi_i) + \sigma_i G_i(s - \xi_i), \right. \\ &\quad \left. X_{p_{i,1}}(\xi_i) + \psi_{s,i} + \left[X'_{p_{i,1}}(\xi_i) + \psi'_{s,i}\right](t - \xi_i) + \sigma_i G_i(t - \xi_i)\right) \\ &= \gamma_{p_{i,1}}(\xi_i, \xi_i) + \sigma_{X_s}^2 + (t + s - 2\xi_i)\gamma'_{p_{i,1}}(\xi_i, \xi_i) + (s - \xi_i)(t - \xi_i) \left(\gamma''_{p_{i,1}}(\xi_i, \xi_i) + \sigma_{X'_s}^2 \right) + \\ &\quad \sigma_i^2 \text{Cov}(G_i(s - \xi), G_i(t - \xi)) \end{aligned} \quad (\text{S.22})$$

and the calculation is very similar for a merging event. The general form for $\gamma_i(s, t)$ is then given

by

$$\gamma_i(s, t) = \begin{cases} \sigma_{X_0}^2 + (s - \xi_i)(t - \xi_i)\sigma_{X'_0}^2 + \sigma_i^2 \text{Cov}(G_i(s - \xi), G_i(t - \xi)) & \text{if } i \in \mathcal{B} \\ \gamma_{p_{i,1}}(\xi_i, \xi_i) + \sigma_{X_s}^2 + (t + s - 2\xi_i)\gamma'_{p_{i,1}}(\xi_i, \xi_i) + \\ (s - \xi_i)(t - \xi_i) \left(\gamma''_{p_{i,1}}(\xi_i, \xi_i) + \sigma_{X'_s}^2 \right) + \sigma_i^2 \text{Cov}(G_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{S} \\ \frac{1}{4} \left(\gamma_{p_{i,1}}(\xi_i, \xi_i) + \gamma_{p_{i,2}}(\xi_i, \xi_i) + 2\gamma_{p_{i,1}, p_{i,2}}(\xi_i, \xi_i) \right) + \sigma_{X_m}^2 + \\ \frac{s+t-2\xi_i}{4} \left(\gamma'_{p_{i,1}}(\xi_i, \xi_i) + \gamma'_{p_{i,2}}(\xi_i, \xi_i) + \gamma'_{p_{i,1}, p_{i,2}}(\xi_i, \xi_i) + \gamma'_{p_{i,2}, p_{i,1}}(\xi_i, \xi_i) \right) + \\ \frac{(s-\xi_i)(t-\xi_i)}{4} \left(\gamma''_{p_{i,1}}(\xi_i, \xi_i) + \gamma''_{p_{i,2}}(\xi_i, \xi_i) + 2\gamma''_{p_{i,1}, p_{i,2}}(\xi_i, \xi_i) \right) + \\ (s - \xi_i)(t - \xi_i)\sigma_{X'_m}^2 + \sigma_i^2 \text{Cov}(G_i(s - \xi), G_i(t - \xi)) & \text{if } i \in \mathcal{M}. \end{cases}$$

Using the same strategy as in (S.21) and (S.22) we can calculate $\gamma'_i(s, t)$ as

$$\gamma'_i(s, t) = \begin{cases} (s - \xi_i)\sigma_{X'_0}^2 + \sigma_i^2 \text{Cov}(G_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{B} \\ \gamma'_{p_{i,1}}(\xi_i, \xi_i) + (s - \xi_i) \left(\gamma''_{p_{i,1}}(\xi_i, \xi_i) + \sigma_{X'_s}^2 \right) + \sigma_i^2 \text{Cov}(G_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{S} \\ \frac{1}{4} \left(\gamma'_{p_{i,1}}(\xi_i, \xi_i) + \gamma'_{p_{i,2}}(\xi_i, \xi_i) + \gamma'_{p_{i,1}, p_{i,2}}(\xi_i, \xi_i) + \gamma'_{p_{i,2}, p_{i,1}}(\xi_i, \xi_i) \right) + \\ \frac{s-\xi_i}{4} \left(\gamma''_{p_{i,1}}(\xi_i, \xi_i) + \gamma''_{p_{i,2}}(\xi_i, \xi_i) + 2\gamma''_{p_{i,1}, p_{i,2}}(\xi_i, \xi_i) \right) + (s - \xi_i)\sigma_{X'_m}^2 + \\ \sigma_i^2 \text{Cov}(G_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{M}. \end{cases}$$

For the IBM,

$$\begin{aligned} \text{Cov}(G_i(s), G'_i(t)) &= E \left\{ \left(\int_0^s B_i(u) du \right) B_i(t) \right\} \\ &= \int_0^s E \{ B_i(u) B_i(t) \} du \\ &= \int_0^s (u \wedge t) du \\ &= \frac{(s \wedge t)^2}{2} + t(s - s \wedge t). \end{aligned}$$

Lastly, we can use the same strategy to calculate $\gamma_i''(s, t)$,

$$\gamma_i(s, t) = \begin{cases} \sigma_{X'_0}^2 + \sigma_i^2 \text{Cov}(G'_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{B} \\ \gamma_{p_{i,1}}''(\xi_i, \xi_i) + \sigma_{X'_s}^2 + \sigma_i^2 \text{Cov}(G'_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{S} \\ \frac{1}{4} \left(\gamma_{p_{i,1}}''(\xi_i, \xi_i) + \gamma_{p_{i,2}}''(\xi_i, \xi_i) + 2\gamma_{p_{i,1}, p_{i,2}}''(\xi_i, \xi_i) \right) + \sigma_{X'_m}^2 + \sigma_i^2 \text{Cov}(G'_i(s - \xi), G'_i(t - \xi)) & \text{if } i \in \mathcal{M} \end{cases}$$

and this completes the description of the covariances.

S.3 Further Details for the Tracking Estimate

In this section we give the details behind the calculation of the conditional density

$$[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z} = z](u, v, p) \quad (\text{S.23})$$

used to achieve our tracking estimate in (9).

In (S.12) of Section S.1 we have written out the density for

$$(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$$

which is the collection of all of the variables in the model of Section 3. Here the script letter denotes the collection of those variables for all targets at all times. For example, $\mathcal{X} = \{(X_i(t_j) : j = 1, \dots, n; i = 1, \dots, m)\}$ is the collection of all x -coordinate values for each target at all times it was observed; see Section S.1.1.3.

Recall that \mathcal{W} is the observability variable and \mathcal{Y} is the y -coordinate. We also let \mathcal{A} denote the collection of all attribute variables we wish to include. For example we might have $\mathcal{A} = (\mathcal{R}_{(1)}, \mathcal{R}_{(2)}, \mathcal{Q}_{(2)}, \mathcal{I})$, which are the smallest radius, largest radius, angle of orientation and intensity for targets respectively. Recall that the variable $N_f(t)$ is the number of false alarms at time t so that $\mathbf{N}_f = (N_f(t_1), \dots, N_f(t_n))$ contains the number of false alarms at each time point. The remaining variables \mathcal{X}_f , \mathcal{Y}_f , and \mathcal{A}_f are the collection of x and y coordinates for false alarms and

attributes for false alarms respectively.

We will use the density for $(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$ to calculate the density in (S.23). Notice that there is a one-to-one mapping

$$g : (\mathcal{P}, \mathcal{Z}) \rightarrow (\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f, \mathcal{Z}).$$

So for a given \mathcal{Z} , the information contained in \mathcal{P} and $(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$ is the same.

Let

$$g^* : (\mathcal{P}, \mathcal{Z}) \rightarrow (\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$$

be the function g without the last variable in its output. Then we can write

$$\begin{aligned} [\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}](u, v, p \mid z) &= P\{\mathcal{U} = u, \mathcal{V} = v, \mathcal{P} = p \mid \mathcal{Z} = z\} \\ &= P\{\mathcal{U} = u, \mathcal{V} = v, (\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f) = g^*(p, z) \mid \mathcal{Z} = z\} \\ &= [\mathcal{U}, \mathcal{V}, (\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f) \mid \mathcal{Z}](u, v, g^*(p, z) \mid z). \end{aligned} \quad (\text{S.24})$$

It is assumed that the distribution of \mathcal{Z} given $(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$ is point uniform on the possible permutations of the values of $(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$ within each time t_j , so

$$[\mathcal{Z} \mid \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](z \mid u, v, w, x, y, a, n_f, x_f, y_f, a_f) = \frac{1}{\prod_{j=1}^n m_j!} I_B(z), \quad (\text{S.25})$$

where

$$B = \{z : g^*(p, z) = (w, x, y, a, n_f, x_f, y_f, a_f) \text{ for some } p\}.$$

So we can calculate the likelihood of $(\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f, \mathcal{Z})$ by multiplying the likelihood given in (S.12) by that in (S.25). To then obtain the density in (S.24), note that for a given value of $\mathcal{Z} = z$,

$$[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f \mid \mathcal{Z}] \propto [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f, \mathcal{Z}]$$

and also realize that for a given z , there are a countable number of arguments

$\alpha_i = (u_i, v_i, w_i, x_i, y_i, a_i, n_{f,i}, x_{f,i}, y_{f,i}, a_{f,i})$ that will make

$$[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f \mid \mathcal{Z}](\alpha_i \mid z) > 0.$$

There is actually a finite number of values of $(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f)$ since they must be a permutation of the values in \mathcal{Z} at each time. But there could be as many as a countable number combinations of births, deaths, splitting and merging events that could be represented by \mathcal{U} and \mathcal{V} . This means that we must have

$$\begin{aligned} [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f, \mid \mathcal{Z}](\alpha_i \mid z) &= \frac{[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f, \mathcal{Z}](\alpha_i, z)}{\sum_{j=1}^{\infty} [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f, \mathcal{Z}](\alpha_j, z)} \\ &= \frac{[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](\alpha_i)}{\sum_{j=1}^{\infty} [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](\alpha_j)}, \end{aligned}$$

where the second equality comes from the fact that the contribution of \mathcal{Z} to the density is a constant by (S.25). Now by equation (S.24) we have

$$[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}](u, v, p \mid z) = \frac{[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](u, v, g^*(p, z))}{\sum_{j=1}^{\infty} [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](u_j, v_j, g^*(p_j, z))},$$

where $\{(u_j, v_j, p_j) : j = 1, 2, \dots\}$ is an enumeration of the possible tracking solutions.

As discussed in Section 4 we also wish to calculate the conditional density of $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ given $\mathcal{Z} = z$ and the event $(\mathcal{U}, \mathcal{V}, \mathcal{P}) \in \mathcal{K}$, where $\mathcal{K} = \{(u_i, v_i, p_i) : i = 1, \dots, K\}$. This is given by

$$[\mathcal{U}, \mathcal{V}, \mathcal{P} \mid \mathcal{Z}, (\mathcal{U}, \mathcal{V}, \mathcal{P}) \in \mathcal{K}](u, v, p \mid z) = \frac{[\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](u, v, g^*(p, z))}{\sum_{j=1}^K [\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathbf{N}_f, \mathcal{X}_f, \mathcal{Y}_f, \mathcal{A}_f](u_j, v_j, g^*(p_j, z))}.$$

S.4 Full Description of the Modified MHT Algorithm

As mentioned in Section 4.3, when we receive a new set of observations, $\mathbf{Z}_j = (Z_1(t_j), \dots, Z_{m_j}(t_j))$, at time t_j we will assume that each observation $Z_i(t_j)$ is either:

1. an observation from an existing target track,
2. the first observation from a target resulting from birth,

3. the first observation from a target resulting from split,
4. the first observation from a target resulting from merger, or
5. a false alarm.

Existing tracks that do not receive a new observation to continue the track at time t_j must either

1. go missing (stay missing), or
2. terminate.

At time t_1 we consider all combinations of each observation treated as an initial target observation or a false alarm. Now assume that we have a set of solutions (hypotheses) for the observations through time t_{j-1} . We then take the new observations, \mathbf{Z}_j , at time t_j and form updated solutions based on all possible combinations of the possibilities listed above. We then hold on to only a subset of these new solutions (those with the highest likelihood) to use to form solutions at the next time step, t_{j+1} . The actual number of solutions to make it through to the next time will vary. Let $\max\{L_j\}$ be the likelihood of the best solution at time t_j . At each time, t_j , we hold on to all solutions that have likelihood greater than $c \max\{L_j\}$ where $c < 1$ is a user-defined parameter. In the interest of speed, we also set a limit, K_s , for the maximum number of solutions that make it through to the next time. The control parameters c and K_s will vary depending on the complexity of the problem. In the problems of Sections 5 and 6 these were set to $c = e^{-10}$ and $K_s = 200$.

Of course it is very inefficient to examine all possible combinations at each time, so we form gates for each of the tracks. A gate is a prediction region for a new observation from a track at time t_j given the previous observations assumed to be part of the track. In Sections 5 and 6 we used a confidence level of $p_g = 0.9999$ for the gate or prediction region. We then limit the possible observations for inclusion into a track to only those that fall into the gate for that track.

We can also do a similar form of gating for observations that we are considering to be the first observations of new tracks resulting from the split of an existing track. We can form a prediction region for $(X_i(t_j), Y_i(t_j)) + (\psi_{X,s}, \psi_{Y,s})$. Recall that $\psi_{X,s}$ is the random error term for the amount the child's x -location will be different from the parent's at the time of split and similarly for $\psi_{Y,s}$. This

can be accomplished by simply adding $\sigma_{X_s}^2$ and $\sigma_{Y_s}^2$ to the x and y components of the conditional variance for the prediction of a new observation in a track. We can then form the prediction region or gate using this inflated variance. We only consider pairs of new observations within this region to possibly be a split pair from the existing track.

For a possible merging event, we can also form a similar region. We can compute the prediction region for the difference between a pair of existing tracks plus a random ψ_d term, for example $(X_1(t_j) - X_2(t_j), Y_1(t_j) - Y_2(t_j)) + (\psi_{X,d}, \psi_{Y,d})$. Recall $\psi_{X,d}$ is the random distance between the parents at the time of a merging event. If the prediction region for this quantity includes zero, we will consider possibility that these targets are the parents in a merging event.

Now suppose targets 1 and 2 can be considered as parents for a possible merging event. We must also find an observation to possibly be the first observation of the track that they merge into. So we must form another prediction region for $1/2(X_1(t_j) + X_2(t_j), Y_1(t_j) + Y_2(t_j)) + (\psi_{X,m}, \psi_{Y,m})$. We would then only consider new observations within this region to be from a new track resulting from the merging of tracks 1 and 2. Note that these prediction regions assume that the merging event takes place at t_j when it really would have taken place at some time in the interval (t_{j-1}, t_j) , but this seems to be adequate provided the time points are not too spread out.

The prediction regions described above can be calculated efficiently via the Kalman Filter by ignoring the dependency resulting from merging and splitting. That is, the first observation is assumed fixed and the others are calculated assuming the model given in (3) and independent of other tracks. It is possible to improve these regions by using the innovations algorithm to compute the conditional distribution of new observations from a track taking into account the previous splitting and merging. This would also be more time consuming however, since the actual covariances take longer to compute than those under the assumed independence of tracks.

In addition to gating, we can usually separate the entire tracking problem into several smaller tracking problems that are “disjoint” from each other. That is, there are often situations where the area (in space and time) that one group of targets occupies does not intersect the area that another group of targets occupies. These groups of observations can be identified with a simple heuristic approach and the algorithm described above can then be applied to each group separately.

We now discuss a way to improve upon the approximate solution provided by the MHT. Suppose we are running the algorithm on a fixed number, n , of time points, and we obtain the set of likely solutions for the last time t_n . Consider the following situation. The solution that would eventually be optimal (have the highest likelihood) at time t_n has a likelihood that is not very high early on in the algorithm when considering only a subset of all the times. We can only hold on to a limited number of possible solutions at each time, so it is possible that the optimal solution will be discarded at an earlier time less than t_n and thus never recovered.

In this case however, the MHT will likely produce a solution that is close to the optimal one. We can improve the set of solutions obtained at the last time, t_n , by a greedy exchange algorithm similar to that described by Sethi & Jain (1987). This basically considers making several simple changes to a solution. If a change results in an increased likelihood, then make the change. This process continues until there are no more beneficial changes to be made.

There are many other changes that we could consider making to improve the performance of the algorithm, but from initial results of the MHT, it seemed to do a very good job of classifying the splitting and merging events correctly according to likelihood, as well as identifying the correct correspondences of observations within tracks. However, where it seemed to struggle the most, was to form short tracks that were made up only of false alarms. This is likely because it had to discard the correct solution, before it realized it would have to pay a penalty when it eventually killed this incorrect track after a short time. In any case, the greedy exchange algorithm we use here only considers the possibility of changing short tracks ($\leq k$ observations) into false alarm observations. For the results in Sections 5 and 6 we set $k = 3$. Considering other possible changes in the greedy exchange step would only improve results.

So for each of the solutions produced when the MHT finishes, we will go through and consider changing any track with less than 4 observations to a collection of false alarm observations. If one of these changes improve the likelihood, then we will keep it.

S.5 Details on Parameter Estimation

In this section we describe how we estimate the parameters of the model given in Section 3. In most cases these estimates are the maximum likelihood estimates (MLE's). In some cases however, the MLE would be too computationally expensive to compute, and we will use other reasonable choices for estimates.

Also for first few time points of the MHT algorithm, some of the estimates given here cannot be computed because there are too few data points. In these cases, we need an initial guess for some of the parameter values. Here we simply used the midpoint (or geometric midpoint for variance parameters) of the parameter limits for an initial guess until enough data was available to estimate these parameters.

S.5.1 Parameters of the Event Model

The Event Model parameters are λ_0 , λ_b , λ_d , λ_s and λ_m . There is also the false alarm rate parameter, λ_f . For the Event Model parameters one can calculate approximate MLE's based on the approximate likelihood given in Section S.1.1.1. The MLE for λ_0 is obviously

$$\hat{\lambda}_0 = N_0,$$

where recall N_0 is the initial number of targets.

Now consider the estimation of the death rate, λ_d . From the approximation in (S.4) we can consider the $U_{d,j}$ for $j = 1, \dots, n$ as independent Poisson observations with parameter $\bar{N}_j \lambda_{d,j} \Delta t_j$. Recall that \bar{N}_j is the average number of targets alive in the interval $[t_j, t_{j+1})$. Denote the collection of \bar{N}_j 's by $\bar{\mathbf{N}}$. Then the contribution to the likelihood in (S.5) from \mathbf{U}_d is

$$[\mathbf{U}_d \mid \bar{\mathbf{N}}](\mathbf{u}) = \prod_{j=1}^n \frac{(\bar{N}_j \lambda_{d,j} \Delta t_j)^{u_j} e^{-\bar{N}_j \lambda_{d,j} \Delta t_j}}{u_j!}.$$

So the derivative of the log likelihood is

$$\frac{d}{d\lambda_d} \log [\mathbf{U}_d \mid \bar{\mathbf{N}}] (\mathbf{u}) = \sum_{j=1}^n \frac{u_j}{\lambda_d} - \bar{N}_j \Delta t_j. \quad (\text{S.26})$$

Setting (S.26) equal to zero gives

$$\hat{\lambda}_d = \frac{\sum_{j=1}^n U_{d,j}}{\sum_{j=1}^n \bar{N}_j \Delta t_j}.$$

In a similar fashion the approximate MLE's of λ_b , λ_s and λ_m can be shown to be

$$\begin{aligned} \hat{\lambda}_b &= \frac{\sum_{j=1}^n U_{b,j}}{\sum_{j=1}^n \Delta t_j} \\ \hat{\lambda}_d &= \frac{\sum_{j=1}^n U_{s,j}}{\sum_{j=1}^n \bar{N}_j \Delta t_j} \\ \hat{\lambda}_d &= \frac{\sum_{j=1}^n U_{m,j}}{\sum_{j=1}^n (\bar{N}_j - 1) \Delta t_j}. \end{aligned}$$

Lastly, consider estimation of the false alarm rate, λ_f . The number of false alarms at each time $N_f(t_j)$ is Poisson with parameter λ_f so the MLE for λ_f is

$$\hat{\lambda}_f = \frac{\sum_{j=1}^n N_f(t_j)}{n}.$$

S.5.2 Parameters of the Observability Model

If we assume the simple *iid* model for missing observations, then the MLE for the observability model parameter, P_d , is the ratio of the number of times the targets were observable to the number of times they existed

$$\hat{P}_d = \frac{\sum_{i=1}^M \sum_{j=1}^n W_i(t_j) I_{[\xi_i, \zeta_i]}(t_j)}{\sum_{i=1}^M \sum_{j=1}^n I_{[\xi_i, \zeta_i]}(t_j)}.$$

S.5.3 Location Parameters

The derivative of the location density is difficult to compute analytically because of the matrix algebra involved. Exact MLE's would then require a time consuming iterative method. We therefore decided to use alternatives to the MLE's for the location parameter estimates. We will present these estimates for the x -coordinate parameters. The estimates for the y -coordinate parameters will be

the same with the obvious notational changes.

S.5.3.1 White Noise Variance

We will first consider estimation of the white noise error variance $\sigma_{X_e}^2$. For the IBM model, the observed location for a path is

$$X_i^*(t_j) = X_i(\xi_i) + tX_i'(\xi_i) + \sigma_i^2 \int_0^{t_j - \xi_i} B_i(s)ds + \varepsilon_{i,j}.$$

So if we make a derivative approximation, we have

$$\frac{D_{i,j}}{\Delta t_j} = \frac{X_i^*(t_{j+1}) - X_i^*(t_j)}{\Delta t_j} \approx X_i'(\xi_i) + \sigma_i^2 B_i(t_j - \xi_i) + \frac{1}{\Delta t_j}(\varepsilon_{i,j+1} - \varepsilon_{i,j}).$$

If we then take the consecutive differences of the $D_j/\Delta t_j$ we have

$$\begin{aligned} D_{i,j}^2 &= \frac{D_{i,j+1}}{\Delta t_{j+1}} - \frac{D_{i,j}}{\Delta t_j} \\ &\approx \sigma_i^2 (B_i(t_{j+1} - \xi_i) - B_i(t_j - \xi_i)) + \frac{1}{\Delta t_j \Delta t_{j+1}} (\Delta t_j \varepsilon_{i,j+2} - (\Delta t_{j+1} + \Delta t_j) \varepsilon_{i,j+1} + \Delta t_{j+1} \varepsilon_{i,j}). \end{aligned}$$

The covariance of consecutive D_j^2 's is given by

$$\text{Cov}(D_j^2, D_{j+1}^2) \approx K_j \sigma_{X_e}^2,$$

where

$$K_j = -\frac{\Delta t_j (\Delta t_{j+1} + \Delta t_{j+2}) + \Delta t_{j+2} (\Delta t_j + \Delta t_{j+1})}{\Delta t_j \Delta t_{j+1}^2 \Delta t_{j+2}}.$$

Hence a method of moments estimate for the measurement error variance is

$$\hat{\sigma}_{X_e}^2 = \frac{1}{N} \sum_{i=1}^M \sum_{j \in O_i} \frac{D_{i,j}^2 D_{i,j+1}^2}{K_j}, \quad (\text{S.27})$$

where O_i is the set of indices, j , that we have four consecutive times $t_j, t_{j+1}, t_{j+2}, t_{j+3}$ where the i^{th} target is observable,

$$O_i = \{j : W_i(t_j) = W_i(t_{j+1}) = W_i(t_{j+2}) = W_i(t_{j+3}) = 1\}$$

and $N = \sum_i n(O_i)$ is the total number of terms in the sum in (S.27).

S.5.3.2 IBM Variance Scalar

For the estimate of the variance scalar σ_i^2 for the i^{th} target, we will make use of the estimate for $\sigma_{X_e}^2$ and use a local linear regression to estimate $X_i(t_j)$'s given the observations $X_i^*(t_j) = X_i(t_j) + \varepsilon_{i,j}$. Once we have an estimate for the $X_i(t_j)$'s, we can form an estimate for σ_i^2 .

The criterion for selection of the bandwidth h will be based on the following rule presented on pages 100-101 of Schimek (2000). Dropping the subscript i , we have n observations $X^*(t_j)$ and we wish to estimate $X(t_j)$. Denote this estimate as $\hat{m}(t_j, h)$. Then as described in Schimek (2000), the prediction risk is

$$E \left[\sum_{j=1}^n (X^*(t_j) - \hat{m}(t_j, h))^2 \right] = E \left[\sum_{j=1}^n (X(t_j) - \hat{m}(t_j, h))^2 \right] + \sigma_{X_e}^2 (n - 2\text{tr}(S)) \quad (\text{S.28})$$

so

$$\frac{1}{n} \sum_{j=1}^n (X^*(t_j) - \hat{m}(t_j, h))^2 \approx \frac{1}{n} \sum_{j=1}^n (X(t_j) - \hat{m}(t_j, h))^2 + \frac{\hat{\sigma}_{X_e}^2}{n} (n - 2\text{tr}(S)).$$

Since it is our goal to minimize the estimation risk which is the first term on the right side of (S.28), we will use the bandwidth, h , that minimizes the quantity

$$R(h) = \frac{1}{n} \sum_{j=1}^n (X^*(t_j) - \hat{m}(t_j, h))^2 - \frac{\hat{\sigma}_{X_e}^2}{n} (n - 2\text{tr}(S)).$$

We will only use this approach to estimate $X_i(t_j)$ if there are more than k observations for the i^{th} path. We set $k = 6$ in practice.

Now we turn to the problem of estimating σ_i^2 . We can do this in the following way. From the

above discussion, we now have an estimate, $\hat{X}_i(t_j)$, for $X_i(t_j)$ and

$$\hat{X}_i(t_j) \approx X_i(0) + tX'_i(0) + \sigma_i^2 G_i(t_j),$$

where $G_i(t)$ is an IBM, $G_i(t) = \int_0^t B_i(s)ds$. The consecutive difference quotient is

$$\frac{\hat{D}_{i,j}}{\Delta t_j} = \frac{\hat{X}_i(t_{j+1}) - \hat{X}_i(t_j)}{\Delta t_j} \approx X'_i(0) + \frac{\sigma_i^2}{\Delta t_j} (G_i(t_{j+1}) - G_i(t_j)).$$

And taking consecutive differences of the $\hat{D}_{i,j}/\Delta t_j$'s gives

$$\hat{D}_{i,j}^2 = \frac{\hat{D}_{i,j+1}}{\Delta t_{j+1}} - \frac{\hat{D}_{i,j}}{\Delta t_j} \approx \frac{\sigma_i^2}{\Delta t_j \Delta t_{j+1}} (\Delta t_j G_i(t_{j+2}) - (\Delta t_j + \Delta t_{j+1}) G_i(t_{j+1}) + \Delta t_{j+1} G_i(t_j)).$$

The variance of $\hat{D}_{i,j}^2$ is then

$$\text{Var}(\hat{D}_{i,j}^2) \approx C_j \sigma_i^2,$$

where C_j is given by

$$C_j = \frac{1}{(\Delta t_j \Delta t_{j+1})^2} \left[\Delta t_j^2 \frac{t_{j+2}^3}{3} + (\Delta t_j + \Delta t_{j+1})^2 \frac{t_{j+1}^3}{3} + \Delta t_{j+1}^2 \frac{t_j^3}{3} - \Delta t_j (\Delta t_j + \Delta t_{j+1}) \left(t_{j+1}^2 t_{j+2} - \frac{t_{j+1}^3}{3} \right) \right. \\ \left. - \Delta t_j \Delta t_{j+1} \left(t_j^2 t_{j+2} - \frac{t_j^3}{3} \right) - \Delta t_{j+1} (\Delta t_j + \Delta t_{j+1}) \left(t_j^2 t_{j+1} - \frac{t_j^3}{3} \right) \right].$$

Hence a method of moments estimate for σ_i^2 is

$$\hat{\sigma}_i^2 = \frac{1}{N} \sum_{j \in O_i} \frac{(D_{i,j}^2)^2}{C_j} \quad (\text{S.29})$$

where here O_i is the set of indices, j , that we have three consecutive times t_j, t_{j+1}, t_{j+2} where the i^{th} target is observable,

$$O_i = \{j : W_i(t_j) = W_i(t_{j+1}) = W_i(t_{j+2}) = 1\}$$

and $N = \sum_i n(O_i)$ is the total number of terms in the sum in (S.29).

Again, we only estimate σ_i^2 in this way if we have greater than $k = 6$ observations for the i^{th} path. If the i^{th} path has less than k observations, then we let σ_i^2 equal the weighted average of the σ_i^2 estimates of the other paths.

S.5.3.3 Initial Conditions Parameters

To estimate the initial conditions parameters, μ_{X_0} , σ_{X_0} , $\mu_{X'_0}$, and $\sigma_{X'_0}$, we will also take advantage of the local regression fits $\hat{X}_i(t)$. We can use the local regression to estimate $X_i(\xi_i)$. Let $t_{i,j}$ be the j^{th} time at which the i^{th} path is observed for $j = 1, \dots, n_i$. We can then estimate $X'_i(\xi_i)$ as

$$\hat{X}'_i(\xi_i) = \frac{\hat{X}_i(t_{i,1}) - \hat{X}_i(\xi_i)}{t_{i,1} - \xi_i}.$$

If the i^{th} path has fewer than $k = 6$ observations, then we can simply let $\hat{X}_i(\xi_i) = X_i(t_{i,1})$ and $\hat{X}'_i(\xi_i) = (X_i(t_{i,2}) - X_i(t_{i,1})) / (t_{i,2} - t_{i,1})$.

Let $\mathcal{B} = \{i : \text{target } i \text{ is a an initial target or a birth}\}$, and let $n(\mathcal{B})$ be the number elements in \mathcal{B} . We can construct estimates for the initial conditions parameters as

$$\begin{aligned} \hat{\mu}_{X_0} &= \frac{1}{n(\mathcal{B})} \sum_{i \in \mathcal{B}} \hat{X}_i(\xi_i) \\ \hat{\sigma}_{X_0}^2 &= \frac{1}{n(\mathcal{B})} \sum_{i \in \mathcal{B}} \left(\hat{X}_i(\xi_i) - \hat{\mu}_{X_0} \right)^2 \\ \hat{\mu}_{X'_0} &= \frac{1}{n(\mathcal{B})} \sum_{i \in \mathcal{B}} \hat{X}'_i(\xi_i) \\ \hat{\sigma}_{X'_0}^2 &= \frac{1}{n(\mathcal{B})} \sum_{i \in \mathcal{B}} \left(\hat{X}'_i(\xi_i) - \hat{\mu}_{X'_0} \right)^2. \end{aligned}$$

S.5.3.4 Splitting and Merging Parameters

Here we will construct estimates for the parameters involved in the initial conditions of splitting or merging events, σ_{X_s} , $\sigma_{X'_s}$, σ_{X_m} , $\sigma_{X'_m}$, and σ_{X_d} . In order to do this we need estimates for $X_i(\zeta_i)$ and $X'_i(\zeta_i)$. We can also use the local regression to estimate $X_i(\zeta_i)$ and in a similar manner we can estimate $X'_i(\zeta_i)$ as

$$\hat{X}'_i(\zeta_i) = \frac{\hat{X}_i(\zeta_i) - \hat{X}_i(t_{i,n})}{\zeta_i - t_{i,n}}.$$

Adopt the convention of Section 3.3 and denote the indices of the parents of target i (if it has any) as $p_{i,1}$ and $p_{i,2}$. Recall that σ_{X_s} is the variance of

$$\psi_{s,i} = X_{p_{i,1}}(\xi_i) - X_i(\xi_i)$$

and $\sigma_{X'_s}$ is the variance of

$$\psi'_{s,i} = X'_{p_{i,1}}(\xi_i) - X'_i(\xi_i)$$

for any path i that is the child of a splitting event. If we let $\mathcal{S} = \{i : \text{target } i \text{ is the child of a splitting event}\}$ and $n(\mathcal{S})$ be the number elements in \mathcal{S} , then we can construct estimates for these parameters as

$$\begin{aligned}\hat{\sigma}_{X_s}^2 &= \frac{1}{n(\mathcal{S})} \sum_{i \in \mathcal{S}} \left(\hat{X}_{p_{i,1}}(\zeta_{p_{i,1}}) - \hat{X}_i(\xi_i) \right)^2 \\ \hat{\sigma}_{X'_s}^2 &= \frac{1}{n(\mathcal{S})} \sum_{i \in \mathcal{S}} \left(\hat{X}'_{p_{i,1}}(\zeta_{p_{i,1}}) - \hat{X}'_i(\xi_i) \right)^2.\end{aligned}$$

Similarly, σ_{X_m} is the variance of

$$\psi_{m,i} = \frac{1}{2}X_{p_{i,1}}(\xi_i) + \frac{1}{2}X_{p_{i,2}}(\xi_i) - X_i(\xi_i)$$

and $\sigma_{X'_m}$ is the variance of

$$\psi'_{m,i} = \frac{1}{2}X'_{p_{i,1}}(\xi_i) + \frac{1}{2}X'_{p_{i,2}}(\xi_i) - X'_i(\xi_i)$$

for any path i that is the child of a merging event. So let $\mathcal{M} = \{i : \text{target } i \text{ is the child of a merging event}\}$ and we can construct estimates of these parameters as

$$\begin{aligned}\hat{\sigma}_{X_m}^2 &= \frac{1}{n(\mathcal{M})} \sum_{i \in \mathcal{M}} \left(\frac{1}{2}\hat{X}_{p_{i,1}}(\zeta_{p_{i,1}}) + \frac{1}{2}\hat{X}_{p_{i,2}}(\zeta_{p_{i,2}}) - \hat{X}_i(\xi_i) \right)^2 \\ \hat{\sigma}_{X'_m}^2 &= \frac{1}{n(\mathcal{M})} \sum_{i \in \mathcal{M}} \left(\frac{1}{2}\hat{X}'_{p_{i,1}}(\zeta_{p_{i,1}}) + \frac{1}{2}\hat{X}'_{p_{i,2}}(\zeta_{p_{i,2}}) - \hat{X}'_i(\xi_i) \right)^2.\end{aligned}$$

Lastly, σ_{X_d} is the variance of

$$\psi_{d,i} = X_{p_{i,1}}(\xi_i) - X_{p_{i,2}}(\xi_i)$$

for any path i that is the child of a merging event. So its estimate is given by

$$\hat{\sigma}_{X_m}^2 = \frac{1}{n(\mathcal{M})} \sum_{i \in \mathcal{M}} \left(\hat{X}_{p_{i,1}}(\zeta_{p_{i,1}}) - \hat{X}_{p_{i,2}}(\zeta_{p_{i,2}}) \right)^2.$$

S.5.4 Size Parameters

Estimation of the size parameters $\mu_{R_{1,i}}$, $\sigma_{R_{1,i}}$, $\mu_{R_{2,i}}$, and $\sigma_{R_{2,i}}$ is complicated by the restriction that mean size must be conserved. Let the size of a target i be defined to be $S_i(t) = R_{1,i}(t)R_{2,i}(t)$ as in Section 3.4. So the constraints are that

$$E(S_i) + E(S_{i+1}) = E(S_{p_{i,1}}) \tag{S.30}$$

if targets i and $i + 1$ are the children of a splitting event and

$$E(S_i) = E(S_{p_{i,1}}) + E(S_{p_{i,2}}) \tag{S.31}$$

if target i is the child of a merging event.

A brief overview of the plan here is to first estimate the mean size for each target, $E(S_i)$, under the constraints above. Then estimate the scale parameter, $\sigma_{S_i}^2$, for S_i . We will use these to obtain an estimate for the shape parameter, μ_{S_i} , of S_i . Lastly, we can then estimate the parameters $\mu_{R_{1,i}}$, $\sigma_{R_{1,i}}$, $\mu_{R_{2,i}}$, and $\sigma_{R_{2,i}}$ by maximum likelihood under the constraints that $\mu_{R_{1,i}} + \mu_{R_{2,i}} = \mu_{S_i}$ and $\sigma_{R_{1,i}}^2 + \sigma_{R_{2,i}}^2 = \sigma_{S_i}^2$. This procedure will ensure that the mean size is conserved by these parameter estimates.

Again let $t_{i,j}$ be the j^{th} time at which the i^{th} path is observed for $j = 1, \dots, n_i$. Notice that for size we do not have the ambiguity problem that can occur with the radii. For example $S = R_1 R_2 = R_{(1)} R_{(2)}$, so estimating the actual parameters of the size, S_i , is not complicated by

only observing the order statistics of the radii. To first estimate the $E(S_i)$, we used a weighted least squares approach. The weights are to be inversely proportional to the sample variance of the observations for S_i . Let $\text{Var}(S_i)$ denote the sample variance of the $S_i(t_{i,j})$ observations for $j = 1, \dots, n_i$. Then we wish to find the values of $E(S_i)$ that minimize

$$\sum_{i=1}^M \sum_{j=1}^{n_i} \frac{1}{\text{Var}(S_i)} \{S_i(t_{i,j}) - E(S_i)\}^2 \quad (\text{S.32})$$

subject to the constraints in (S.30) and (S.31). This is carried out using the Lagrangian Multiplier method. Denote the resulting minimizers of expression (S.32) as $\hat{E}(S_i)$.

We will then estimate the scale parameter for S_i , $\sigma_{S_i}^2 = \sigma_{R_{1,i}}^2 + \sigma_{R_{2,i}}^2$ by the unconstrained MLE. This is just the sample variance of the $\log(S_i(t_{i,j}))$ observations for $j = 1, \dots, n_i$. Denote this estimate as $\hat{\sigma}_{S_i}^2$. Notice that since S_i is log-normal

$$E(S_i) = e^{\mu_{S_i} + \frac{1}{2}\sigma_{S_i}^2},$$

where $\mu_{S_i} = \mu_{R_{1,i}} + \mu_{R_{2,i}}$ is the shape parameter of S_i . So once the estimates $\hat{E}(S_i)$ and $\hat{\sigma}_{S_i}^2$ are obtained, we can let

$$\hat{\mu}_{S_i} = \log\{\hat{E}(S_i)\} - \frac{1}{2}\hat{\sigma}_{S_i}^2.$$

Finally, we can estimate the parameters $\mu_{R_{1,i}}$, $\sigma_{R_{1,i}}$, $\mu_{R_{2,i}}$ and $\sigma_{R_{2,i}}$ by maximum likelihood under the constraints that $\hat{\mu}_{R_{1,i}} + \hat{\mu}_{R_{2,i}} = \hat{\mu}_{S_i}$ and $\hat{\sigma}_{R_{1,i}}^2 + \hat{\sigma}_{R_{2,i}}^2 = \hat{\sigma}_{S_i}^2$. If we set $\mu_{R_{2,i}} = \mu_{S_i} - \mu_{R_{1,i}}$ and $\sigma_{R_{2,i}}^2 = \sigma_{S_i}^2 - \sigma_{R_{1,i}}^2$, this is equivalent to the estimation of $\mu_{R_{1,i}}$ and $\sigma_{R_{1,i}}$ with $\mu_{R_{1,i}}$ unconstrained and $\sigma_{R_{1,i}}$ confined to the interval $(0, \hat{\sigma}_{S_i}^2)$. Recall from equation (S.13) that this likelihood is a product of sums, and we will therefore need an iterative method to maximize it. Thus this estimation is carried out using a Newton Raphson algorithm. Notice however that this is only a two dimensional maximization and we can use the unconstrained MLE's assuming $R_1 = R_{(1)}$ for the parameters as starting points. The optimization can therefore be carried out quite quickly. This is the reason we chose to first reduce the problem to a two dimensional estimation for each target instead of applying a Newton Raphson approach to the entire problem to begin with.

S.5.5 Orientation Parameters

For the estimation of the angle of orientation parameters, α_i and β_i , we again use maximum likelihood. Recall from (S.14) and (S.15) that the likelihood for the $Q_i(t_j)$ depends on the $R_{(1),i}(t_j)$, $R_{(2),i}(t_j)$ and their corresponding parameters $\mu_{R_{1,i}}$, $\sigma_{R_{1,i}}$, $\mu_{R_{2,i}}$, and $\sigma_{R_{2,i}}$. So we can substitute the parameter estimates $\hat{\mu}_{R_{1,i}}$, $\hat{\sigma}_{R_{1,i}}$, $\hat{\mu}_{R_{2,i}}$, and $\hat{\sigma}_{R_{2,i}}$ from Section S.5.4 into the density for \mathcal{Q} given in (S.16). We then again use Newton Raphson to find the values of α_i and β_i that maximize the likelihood given in (S.16).

S.6 Detailed Simulation Results

In this section, we present some results of the tracking algorithm on simulated data. For all of these simulations, the data, \mathcal{Z} , is assumed to come from the model given in Section 3. The random motion component, $G_i(t)$ is an integrated Brownian Motion for all targets. The parameters used to simulate the different cases will be given below. All of the simulations use common location parameters. These values were meant to make the target tracks produced from the model behave like the storm tracks of Section 6. So in all of the realizations we set, $\mu_{X_0} = -113$, $\sigma_{X_0}^2 = 100$, $\mu_{X'_0} = 1.5$, $\sigma_{X'_0}^2 = .1$, $\sigma_i^2 = 0.1$ for all i , $\sigma_{X_s}^2 = .5$, $\sigma_{X'_s}^2 = .01$, $\sigma_{X_m}^2 = .125$, $\sigma_{X'_m}^2 = .01$, $\sigma_{X_d}^2 = 1$, $\sigma_{X_e}^2 = 0$, $\mu_{Y_0} = 37.5$, $\sigma_{Y_0}^2 = 100$, $\mu_{Y'_0} = 0$, $\sigma_{Y'_0}^2 = 2$, $\eta_i^2 = .1$ for all i , $\sigma_{Y_s}^2 = .5$, $\sigma_{Y'_s}^2 = .5$, $\sigma_{Y_m}^2 = .125$, $\sigma_{Y'_m}^2 = .01$, $\sigma_{Y_d}^2 = 1$, and $\sigma_{Y_e}^2 = 0$, where $\mu_{X_0}, \sigma_{X_0}^2, \dots, \sigma_{X_e}^2$ are defined in Section 3.3. The parameters $\mu_{Y_0}, \sigma_{Y_0}^2, \dots, \sigma_{Y_e}^2$ are the counterparts for the y -coordinate. Also σ_i^2 and η_i^2 are the variance scalars multiplied to $G_i(t)$ in equations (3), (5), and (7) for $X_i(t)$ and $Y_i(t)$ respectively.

All of these simulations allow for false alarms to appear at each time with rate $\lambda_f = 8.0$ so we can expect about 8 false alarms at each time. We also set the probability of detection $P_d = 0.95$. The parameters λ_0 , λ_b , λ_d , and λ_s , and λ_m are different for each simulation and will be described for each case.

For the parameter estimation, we restricted the parameter values to the followings sets $\lambda_0 \in [0, 25]$, $\lambda_f \in [0, 25]$, $\lambda_b \in [0.001, .25]$, $\lambda_d \in [0.001, .15]$, $\lambda_s \in [0.001, .15]$, $\lambda_m \in [0.001, .15]$, $P_d \in [0.5, 1.0]$, $\mu_{X_0} \in [-120, -85]$, $\sigma_{X_0}^2 \in [500, 1000]$, $\mu_{X'_0} \in [0, 5]$, $\sigma_{X'_0}^2 \in [0.001, 5.0]$, $\sigma_i^2 \in [0.001, 10.0]$, $\sigma_{X_s}^2 \in [0.001, 1.5]$, $\sigma_{X'_s}^2 \in [0.0, 1.0]$, $\sigma_{X_m}^2 \in [0.001, 0.5]$, $\sigma_{X'_m}^2 \in [0.0, 1.0]$, $\sigma_{X_d}^2 \in [0.001, 5.0]$, $\sigma_{X_e}^2 \in$

$[0.0, 1.0]$, $\mu_{Y_0} \in [25, 50]$, $\sigma_{Y_0}^2 \in [500, 1000]$, $\mu_{Y'_0} \in [-5, 5]$, $\sigma_{Y'_0}^2 \in [0.5, 10.0]$, $\eta_i^2 \in [0.001, 10.0]$, $\sigma_{Y_s}^2 \in [0.001, 1.5]$, $\sigma_{Y'_s}^2 \in [0.0, 1.0]$, $\sigma_{Y_m}^2 \in [0.001, 0.5]$, $\sigma_{Y'_m}^2 \in [0.0, 1.0]$, $\sigma_{Y_d}^2 \in [0.001, 5.0]$, and $\sigma_{Y_e}^2 \in [0.0, 1.0]$.

There are six cases that we considered here:

- (i) **Birth only** For this simulation we set $\lambda_0 = 2.0$, $\lambda_b = 0.20$ so that we would have an average of approximately 2 births in a time interval $[0, 9]$. We then set $\lambda_d = \lambda_s = \lambda_m = 0$ so we could isolate the tracking algorithm's ability to identify birth events. We also restricted the simulation to the set of realizations that have at least one birth event.
- (ii) **Death only** In these simulations, we set $\lambda_0 = 4.0$, $\lambda_d = 0.10$. This makes for an average of about 2.5 deaths in the time interval and we restricted our focus to the set of realizations that had at least one death. We then set $\lambda_b = \lambda_s = \lambda_m = 0$.
- (iii) **Splitting only** In the splitting only simulations, we forced there to be exactly one target that split into two targets at a random uniformly distributed time in the interval $(1.0, 8.0)$.
- (iv) **Merging only** In a similar manner to the splitting only simulations, the merging only simulations, have exactly one merger by two targets at a uniformly distributed time in the interval $(1.0, 8.0)$.
- (v) **Completely Random** These are completely unrestricted realizations from the model with event parameters set as $\lambda_0 = 4$, $\lambda_b = 0.1$, $\lambda_d = .02$, $\lambda_s = 0.06$, and $\lambda_m = .08$.
- (vi) **Completely Random w/ Size** These are the same realizations as in case (v) but now with size information to be used in the tracking algorithm. The radius variables $R_{(1)}$ and $R_{(2)}$ are being used along with location here to compute the likelihood.

For each realization we would generate two random variables $z_1 \sim \mathcal{N}(0.6, .01)$, $z_2 \sim \mathcal{N}(0.8, .01)$ and set $\mu_{R_{1,i}} = z_1 \wedge z_2$, $\mu_{R_{2,i}} = z_1 \vee z_2$. We then set the log-normal scale parameters, $\sigma_{R_{1,i}}^2 = \sigma_{R_{2,i}}^2 = 0.025$ for all i . In the parameter estimation, parameter limits for size were set for $\mu_{S,i} = \mu_{R_{1,i}} + \mu_{R_{2,i}}$ and $\sigma_{S,i}^2 = \sigma_{R_{1,i}}^2 + \sigma_{R_{2,i}}^2$. The parameter limits for $\mu_{S,i}$ were set to be the min and max of the observed values of the log sizes, $\mu_{S,i} \in [\min\{\log(S_{i,j})\}, \max\{\log(S_{i,j})\}]$

and $\sigma_{S,i}^2 \in [0.001, 1.0]$. Also recall that $\mu_{S,i}$ is also restricted by merging and splitting so that the mean size of the parent(s) adds to the mean size of the child(ren). The radius parameters were otherwise free in the maximum likelihood estimation.

We set the false alarm size parameters to $\mu_{R_{1,f}} = 0.00$, $\mu_{R_{2,f}} = 0.25$, and $\sigma_{R_{1,f}}^2 = \sigma_{R_{2,f}}^2 = 0.25$. This produces false alarms that are smaller than targets on average, but possibly similar in size to small or medium size targets.

For each case we generated $N = 100$ realizations. These simulations take place on the time interval $[0, 9]$ with $\Delta t_j = 1$ for all j so that $\mathbf{t} = (0, 1, \dots, 9)$. An example of a realization from the completely random (CR) model was given in Figure 5 in the main article. We wish to investigate the same hypotheses 1-4, posed in the previous section now with the presence of clutter (i.e., false alarms).

In these simulations we have the following hypotheses we wish to investigate.

1. The percentage of births, deaths, splits, and mergers labeled correctly in each of the first four simulations respectively, will be roughly equal to the rates of correctly labeled events in the full model realizations of simulation (v).
2. Since birth is symmetric to death in reverse time, we would expect that the rate of correctly labeled births would be similar to that of correctly labeled deaths.
3. Since also splitting is symmetric to merging in reverse time, we would expect that the rate of correctly labeling these two events would be similar.
4. The results with additional size information in simulation (vi) should be an improvement over those in simulation (v).

S.6.1 Simulation Results for Cases (i)-(vi)

The simulation results of each of the six cases are given as the columns of Table S.1. In the following we describe each of the summary statistics that make up the rows of Table S.1.

% Best Est Correct This is the percentage of times that $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ from (12) was equal to the correct solution $(\mathcal{U}, \mathcal{V}, \mathcal{P})$.

	Birth	Death	Split	Merge	CR	CR w/Size
% Best Est Correct	72.0	60.0	61.0	82.0	67.0	92.0
% Births Correct	94.8	-	-	-	83.1	100.0
% Deaths Correct	-	79.2	-	-	70.2	95.7
% Splits Correct	-	-	93.0	-	87.0	100.0
% Mergers Correct	-	-	-	97.0	90.7	98.7
% Targets Correct	99.3	99.0	98.9	99.9	99.0	99.8
% FAs Correct	97.4	95.2	97.4	99.5	99.2	99.6
% Falling in 95% CS	94.0	79.0	86.0	98.0	81.0	96.0
(5%)	0.015	0.000	0.001	0.059	0.000	0.068
Prob of True (25%)	0.292	0.090	0.074	0.598	0.207	0.996
(50%)	0.850	0.685	0.746	0.947	0.996	0.996
(5%)	0.315	0.209	0.294	0.363	0.265	0.790
Prob of Best Est (25%)	0.614	0.508	0.583	0.695	0.582	0.996
(50%)	0.906	0.877	0.891	0.947	0.996	0.996
(5%)	0.881	0.610	0.831	0.938	0.856	1.000
Track Purity (25%)	1.000	1.000	1.000	1.000	0.956	1.000
(50%)	1.000	1.000	1.000	1.000	1.000	1.000
(5%)	0.903	1.000	0.997	1.000	0.869	1.000
Prob of Target (25%)	1.000	1.000	1.000	1.000	1.000	1.000
(50%)	1.000	1.000	1.000	1.000	1.000	1.000
(5%)	0.922	0.705	0.897	0.922	1.000	1.000
Prob of FA (25%)	1.000	1.000	1.000	1.000	1.000	1.000
(50%)	1.000	1.000	1.000	1.000	1.000	1.000

Table S.1: Results of 100 Realizations With Clutter

% Births Correct Percentage of all birth events in the simulation that were labeled correctly by the estimate, $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$.

% Deaths Correct Percentage of all death events in the simulation that were labeled correctly by the estimate.

% Splits Correct Percentage of all splitting events in the simulation that were labeled correctly by the estimate.

% Mergers Correct Percentage of all merging events in the simulation that were labeled correctly by the estimate.

% Falling in 95% CS We form a 95% confidence set of solutions for each realization. This is the percentage of times that the 95% confidence set contained the correct solution.

Prob of True This is the estimated posterior probability that the correct solution, $(\mathcal{U}, \mathcal{V}, \mathcal{P})$, is correct calculated using (13). These three rows are respective quantiles from the 100 realizations for these probabilities

Prob of Best Est This is the estimated probability that the estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ in (9) is correct given the data, again presented by the quantiles.

Track Purity These three rows are quantiles for the overall track purity for each realization. The overall track purity is defined in the paragraph below.

% Targets Correct This is the percentage of all targets at all times in the simulation that were labeled correctly as targets by the estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$.

% FAs Correct Percentage of all false alarms in the simulation that were labeled correctly by the estimate.

Prob of Target This is the probability given the data that a given target at the last time step should be labeled a target. The three rows are the quantiles of these probabilities over all of the targets in the last time step in all of the realizations.

Prob of FA This is the same as “Prob of Target” only for false alarms.

We present a definition of track purity that is slightly different than that given by Mori, Chang, Chong & Dunn (1986). In the correct solution, $(\mathcal{U}, \mathcal{V}, \mathcal{P})$, consider a given track i composed of observations produced by target i . Of all the tracks that make up the estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$, find the track i' that contains the most observations in common with track i in $(\mathcal{U}, \mathcal{V}, \mathcal{P})$. The track purity for track i is defined to be the proportion of the observations that make up track i that are also part of track i' in the estimate $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$. The overall track purity is then the weighted average (by number of observations in the track) of individual track purities.

For example if $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ had two tracks; track 1 with 5 observations and track 2 with 10 observations. And the estimate, $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$, has three tracks; track 1, track 2, and track 3. Where track 1 in $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ is identical to track 1 in $(\mathcal{U}, \mathcal{V}, \mathcal{P})$. Track 2 in $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ is the first 7 observations of track 2 in $(\mathcal{U}, \mathcal{V}, \mathcal{P})$ and track 3 in $(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \hat{\mathcal{P}})$ is the last 3 observations of track 2 in $(\mathcal{U}, \mathcal{V}, \mathcal{P})$.

Then the track purity for track 1 is 1.0. The track purity for track 2 is 0.7 and the overall track purity is $[5(1.0) + 10(0.7)]/15 = 0.8$.

Now refer back to the four hypotheses we posed earlier. Recall that the first hypothesis states that the first four simpler simulations will translate their error rates to the more complicated CR model case. From Table S.1 we can see that the percentage of births labeled correctly in the birth only simulation (94.8%) is somewhat higher than that in the CR model (83.1%). The percentage of deaths labeled correctly in the death only case (79.2%) is also a bit higher than that for the CR model (70.2%). The percentage of splits correct in the splitting only case is closer to the CR model, (93.0%) versus (87.0%). Lastly the percentage of mergers correct in the merging only case is also slightly higher than in the CR model, (97.0%) to (90.7%). So it appears that these rates for the first four simulations are in general a little bit higher than those for the CR model simulation.

For the second hypothesis, the percentage of births correct in the birth only case (94.8%) is again a bit higher than the percentage of deaths correct in the death only case (79.2%). This is again likely due to deaths near the end of the time window and high death rate resulting in shorter tracks. In fact, the fifth percentile for track purity in the death only case is only 0.610 here which leads us to believe there are a few instances where the algorithm decided to label a short track as clutter instead of paying the price for a death.

For the third hypothesis, the percentage of splits correct in the splitting only case (93.0%) is again quite similar to the percentage of mergers correct in the merging only case (97.0%). So again there is a good indication that third hypothesis is correct.

Recall that the last hypothesis says that the size information will improve the results. This was not abundantly clear in the simulations without clutter. However, in the presence of clutter, the size information adds quite a bit of discernment power. The percentage of correct estimates jumps from 67.0% for the CR model without size to 92.0% for the CR model with size. Also if we look at the probability given the data that the correct solution is correct we see that these are substantially higher when we include size. Lastly, the coverage of the 95% confidence sets is significantly improved from 81.0%, to 96.0% when we use size in the algorithm.

The coverage of the 95% confidence sets for these simulations (94.0%, 79.0%, 86.0%, 98.0%, 81.0%, and 96.0% for simulations 1-6 respectively) dropped off some from the simulations without clutter. One explanation for this, referring back to (10), is that these sets assume that the correct answer is in the collection of solutions we obtained from the MHT algorithm. If it is not always in this collection, then of course our distribution of solutions given in (10) will not be correct. Also, since we estimate parameters for each of the possible solutions, this also introduces some bias. Overall though, these confidence sets and probabilities provide us with at least a rough guide as to how confident we should be in the estimated solution(s).

Notice that although the estimate is not always the correct solution for these simulations, the track purity values are always high. Only 5% of track purities for any of the cases was below 0.88 with the exception of the death only simulation which had 5% below 0.610. The percentages of Targets correct and false alarms correct were also uniformly high. These were usually around 99% for most cases and never lower than 95.2% for any of the simulations.

S.6.2 Decreasing Time Increments

The set of simulations considered in the section uses a model identical to that of the CR model realizations with clutter of Section S.6.1. Here however, we use three different time increments, $\Delta t = 1.0$, $\Delta t = 0.5$, and $\Delta t = 0.1$. The conjecture here is that there is a convergence of the estimate to the correct solution as the time increment approaches zero.

From Table S.2 we can see that the estimation does improve substantially as Δt becomes smaller. We see a dramatic improvement in the number of correct estimates. The percentage goes from 67.0% for the $\Delta t = 1.0$ case, to 79.0% for the $\Delta t = 0.5$ case, to 99.0% for the $\Delta t = 0.1$ case. Also for the probability of the correct solution given the data, 25% of the $\Delta t = 1.0$ probabilities are less than 0.207, but only 5% of the $\Delta t = 0.1$ probabilities are less than 0.834. It appears as though there is a convergence of this estimate to the correct solution.

	CR, $\Delta t=1.0$	CR, $\Delta t=0.5$	CR, $\Delta t=0.1$
% Best Est Correct	67.0	79.0	99.0
% Births Correct	83.1	88.2	100.0
% Deaths Correct	70.2	91.1	98.2
% Splits Correct	87.0	95.7	100.0
% Mergers Correct	90.7	97.7	100.0
% Targets Correct	99.0	99.7	100.0
% FAs Correct	99.2	99.9	100.0
% Falling in 95% CS	81.0	90.0	100.0
(5%)	0.000	0.006	0.834
Prob of True (25%)	0.207	0.622	0.994
(50%)	0.996	0.970	0.996
(5%)	0.265	0.499	0.836
Prob of Best Est (25%)	0.582	0.803	0.994
(50%)	0.996	0.974	0.996
(5%)	0.856	0.940	1.000
Track Purity (25%)	0.956	1.000	1.000
(50%)	1.000	1.000	1.000
(5%)	0.869	0.999	1.000
Prob of Target (25%)	1.000	1.000	1.000
(50%)	1.000	1.000	1.000
(5%)	1.000	1.000	1.000
Prob of FA (25%)	1.000	1.000	1.000
(50%)	1.000	1.000	1.000

Table S.2: Results of 100 Realizations With Decreasing Δt

S.7 Detection Algorithm

The problem of target or object identification in images has been studied quite thoroughly. It is not our goal to make a contribution in this area, hence a detailed description of these techniques will not be given. We simply describe the details of the particular identification technique we chose to use on the storm tracking problem. For a good summary of other imaging techniques, see Rosenfeld & Kak (1982).

Recall, the goal of the detection algorithm is to go through each image and record the location of each target (storm) that it finds. In our case, we will record the size and orientation of the storms as well.

An image consists of intensity values $I_{i,j}$ for each of the pixels. We start by thresholding the intensities at a value α . At this point, all pixels with intensities $I_{i,j} < \alpha$ will be set to zero. We

then consider all of the pixels with $I_{i,j} > \alpha$ and we wish to group these pixels together to make up the targets.

Simply stated, all pixels with $I_{i,j} > \alpha$ that are “connected” to each other are part of the same target. There are two common definitions of connected pixels. Two pixels are *4-connected* if they share one of their 4 sides with each other. Two pixels are *8-connected* if they share a common side or corner. We have found that the 4-connected definition works well for the storms problem, but certainly the best one to use is problem dependent.

We now have a collection of targets, defined by their corresponding cluster of pixels. To specify location, size and orientation of the targets, we fit an ellipse to each target (cluster of pixels). This can be accomplished by estimating a bivariate Gaussian distribution for each target and using the 99% contour of the density.

The mean and covariance of the Gaussian distribution used to fit an ellipse to a given target are given by the following. Suppose $x_{i,j}$, $y_{i,j}$, are the coordinates of the center of pixel i,j . The moments for a given target are given by

$$\hat{\mu}_x = \frac{\sum I_{i,j} x_{i,j}}{\sum I_{i,j}}$$

and similarly for $\hat{\mu}_y$, $\hat{\sigma}_x$, $\hat{\sigma}_y$, and $\hat{\sigma}_{xy}$ where the sum is taken over the pixels (i,j) , that compose that target.

The location of the target is then given by $(\hat{\mu}_x, \hat{\mu}_y)$. The length of the radii $R_{(1)}$ and $R_{(2)}$ of the target are given by the minor and major axes of the 99% contour ellipse. The angle of orientation $Q_{(2)}$ is also obtained from the ellipse. Refer to Figure 4 for an illustration of this.

For this application, the pixel intensities ranged anywhere from 0.00 to 150.00 mm/hour of rainfall which roughly equates to 0.0 to 6.0 inches of rain per hour. Most pixels that made up storms had intensities between 1.00 and 10.00 mm/hour. We used a threshold of $\alpha = 0.10$ with the 4-connected definition. In addition, we are only considering mesoscale systems here, which are storms with $R_2 > 1^\circ$. All other storms are discarded, so this could be considered a second stage of thresholding.